PL ISSN 0860-1674
Nr indeksu 353965

10

Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"

NR 10 PAZDZIERNIK 1986 CENA 100 ZŁ

2098

9170
9170
9170
9170
9170



PASCAL cz. I

SPRZĘŻENIE WEMNE

Jaki jest podstawowy warunek uzyskania pożytków z informatyki? Opracowany przez MNSzWiT dokument pt. „Program rozwoju zastosowań techniki komputerowej w procesach kształcenia w szkołach wyższych w latach 1985–1990" przynosi na to pytanie odpowiedź zdecydowaną i klarowną. Brzmi ona: „Upowszechnienie kształcenia informatycznego przez szkoły wyższe jest jedyną drogą do efektywnego i skutecznego rozwoju informatyki i jej zastosowań w gospodarce narodowej". Koniec, kropka, a co najważniejsze — prawda. Tylko co z tej prawdy wynika?

W tym miejscu chciałbym poczynić dygresję osobistą. Mniej więcej 15 lat temu opublikowałem w Tygodniku ITD swój pierwszy w życiu artykuł. Z pozycji studenta trzeciego roku Wydziału Elektroniki PW kreśliłem wizję rozwoju elektronicznej techniki obliczeniowej w Polsce. Artykuł nosił tytuł: „Kogo uczyć informatyki?", a odpowiedź stwierdzała, a jakże, że wszystkich, tzn wszystkich studentów wszystkich wydziałów wszystkich typów uczelni. Było to dla nas na elektronice tak wówczas oczywiste, jak to że dwa razy dwa jest zazwyczaj cztery. Byliśmy zresztą przekonani, iż za kilka lat taka powszechna edukacja z pewnością stanie się faktem.

Minęło 15 lat i oto z diagnozy zawartej we wspomnianym „Programie rozwoju" można się dowiedzieć, że:

W ciągu minionego dziesięciolecia nie udało się rozwinąć problemu wyposażenia uczelni we współczesny sprzęt komputerowy.

Dotychczasowe programy studiów na różnych kierunkach (wyłączając informatykę) w większości przypadków zawierają tylko informacje o zakresie nauczania podstaw informatyki, natomiast nie przewidują zastosowania jej metod w przedmiotach zawodowych, w projektach, pracach przejściowych i dyplomowych...

Na kierunkach informatycznych powstały takie niekorzystne zjawiska jak zahamowanie rozwoju kadr naukowych, niedostateczna w stosunku do potrzeb liczba hablilitacji i doktoratów, opóźnienie tematyki badawczej ośrodków naukowych w stosunku do kierunku badań informatycznych na świecie...

Brak sprzętu, oprogramowanie i kadry wykładowców sprawia, że kształcenie informatyczne na kierunkach nieinformatycznych jest niedostateczne, a co za tym idzie umiejętności absolwentów polskich szkół wyższych w zakresie programowania oraz czynnego posługiwania się środkami i metodami informatyki są niewystarczające.

Wszystko to jest alarmujące, ale najgroźniejszy efekt daje dopiero zsumowanie tych trudności cząstkowych. Okazuje się otóż, że nastąpiło swoiste ujemne sprzężenie zwrotne, tzn. „Brak dostatecznej liczby wykładowców i nauczycieli informatyki jest skutkiem tych trudności, ale sam z kolei staje się powodem ograniczania zakresu i poziomu kształcenia i szkolenia informatycznego a przez to dalszego narastania negatywnych zjawisk w informatyzacji kraju".

Jak przerwać tę pętlę niemocy? Autorzy Programu widzą dwa główne skądinąd oczywiste sposoby. Należy zapewnić sprzęt i ludzi. Ale jak sami zaraz stwierdzają: „Sprawa sprzętu budzi największy niepokój". Żeby niepokój ten przezwyciężyć konieczne są nakłady w wysokości 23 mld zł i 15 mln dolarów, wtedy nasze uczelnie można będzie w miarę zadawalająco nasycić sprzętem komputerowym. Podaję te liczby gwoli dziennikarskiego obowiązku, bo jest to jedyne, co w przypadku takich sum mogę zrobić.

Mogę natomiast apelować o zwiększenie liczby kształconych informatyków, stanowiących przecież kadrę niezbędną dla podejmowania jakichkolwiek działań w tym zakresie. Obecnie rozpoczyna corocznie studiowanie informatyki ok. 300 osób w skali kraju. Projekt planu na lata 1986–90 mówi o 380. Tymczasem autorzy „Programu..." przekonują, że należy corocznie przyjmować na studia informatyczne minimum 520 osób (tj. tyle co w roku 1980). Należy poprzeć te postulaty, gdyż inaczej możemy stanąć przed najtrudniejszą barierą rozwoju — barierą kadr.

Publikujemy obok rozmowę z Norbertem Krzakiem, studentem pierwszego roku Instytutu Informatyki PW (na egzaminie wstępnym uzyskał 196 punktów na 200 możliwych). Składając na ręce Norberta życzenia sukcesów adresowane do wszystkich rozpoczynających w październiku br. swą akademicką przygodę chciałbym w imieniu całego Zespołu „Bajtka" wyrazić nadzieję, że to właśnie wy przyczynicie się do przerwania pętli niemocy polskiej informatyki. Nadzieje te adresuję zresztą nie tylko do studentów informatyki.

Informatykę można bowiem w dużym stopniu porównać z matematyką, której naucza się przecież na większości kierunków studiów. Jednak wykłady z matematyki straciłyby swój sens, gdyby wykładowcy przedmiotów zawodowych nie korzystali z metod matematycznych. Tak samo wykłady z podstaw informatyki tracą swój sens, jeżeli wykładowcy z przedmiotów zawodowych nie będą korzystać z metod i środków informatyki. Ale... wykładowcy nie są wcale niezbędni, aby mogli się posługiwać informatyką studenci! Co stoi bowiem na przeszkodzie, aby ułatwić sobie na przykład wykonywanie standardowych przecież projektów prac przejściowych, poprzez ułożenie programu, który zaoszczędzi całej grupie kilku tygodni mechanicznych obliczeń i kreśleń. Że zakpicie sobie w ten sposób z paru asystentów? A czy ktoś tego zabronił?

Zakładajcie studenckie koła naukowe informatyki, opanowujcie istniejące kluby komputerowe organizowane pod patronatem różnych instytucji i organizacji (które już wyciągnęły z państwowej kiesy sporo pieniążków na ten cel), domagajcie się szerszego otwarcia drzwi do wydziałowych i uczelnianych ośrodków obliczeniowych, w ostateczności wpadnijcie do „Bajtka". Na sprzęcie, który posiadamy da się obliczyć i wykreślić niezłe prace przejściowe? Róbcie zresztą cokolwiek. Tylko nie czekajcie z założonymi rękami aż informatyka w Polsce stanie się faktem!

Waldemar Siwiński



WYBIERZ SAM

GRA O JUTRO
Z dwóch stron katedry 3
SWEGO NIE ZNACIE
Polski IBM 5
Nie tylko małe 5
PROGRAMOWAĆ MOŻE KAŻDY
Programujemy w PASCAL-u cz.I . 7
KLAN SPECTRUM
Katalog 8
Grafika 9
Pióro świetlne 10
BETA BASIC 11
KLAN ATARI
Zegar 12
KLAN AMSTRAD — SCHNEIDER
Sprawdź refleks 14
Muzyka robota 14
Transmat 15
CO JEST GRANE
MOVIE 15–17
Kennedy Approach 18
Nonterraqueous 19
Chuckie Egg 19
KLAN COMMODORE
Polski alfabet cz.I 20
Tajemnice C-128 21
Poradnik młodego pirata 22
KLAN NIETYPOWYCH
Sharp 23
JAK TO ROBIĄ INNI
Szkoła nr 1170 24
PRZED EKRANEM
Dziś piszemy książkę 25
SAMI O SOBIE
O złotą dyskietkę Bajtka 26
Informik 26
Horacy 26
Indywidualny Bank Danych 28
GIEŁDA 30
TYLKO DLA PRZEDSZKOLAKÓW
Alibaba 31
NIE TYLKO KOMPUTERY
Kryptonim „Fobos" 32

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH".

ADRES: 00-685 Warszawa, ul Wspólna 61. Telefon 21-12-05.

Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych".

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu „Bajtka"), Roman Poznański (z-ca sekretarza redakcji „SM" — sekretarz zespołu „Bajtka"), Krzysztof Czernek, Klaudiusz Dybowski, Sławomir Gajda (red. techniczny), Andrzej Kowalewski, Wiesław Migut, Sławomir Polak, Tomasz Pyć, Wanda Roszkowska (opr. graficzne), Michał Silski, Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski, Sergiusz Wolicki. Zdjęcia w numerze: Leopold Dzikowski.

WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.

Cena 100 zł.

Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.

Zam. nr 1214/86, nakład 250 000 egz. P 101



Chcę zostać dobrym inżynierem

Im więcej osób dobija się na nasz kierunek — — tym więcej można wybrać naprawdę zdolnych.

Z DWÓCH STRON KATEDRY

Rozmowa z Norbertem Krzakiem

— beanem Wydziału Elektroniki Politechniki Warszawskiej — kierunek informatyka.

— Na dwieście punktów możliwych do zdobycia uzyskałeś podczas egzaminów wstępnych aż 196, z egzaminów kierunkowych dostałeś same piątki. Czy egzamin był dla ciebie trudny?

— Nie, chociaż był dość obszerny. Na przykład pisemny z matematyki składał się z dwudziestu pytań, obejmował wszystkie działy matematyki, które poznaliśmy w szkole, ale wymagana wiedza nie wykraczała poza program. Dziesięć łatwych pytań po dwa punkty i dziesięć trochę trudniejszych — po trzy punkty. Podobnie z fizyki było dwadzieścia pytań. Na każdy test dostaliśmy trzy godziny. W sumie egzaminy trwały pięć dni.

— Czy egzamin powinien zawierać pytania z informatyki?

— Nie, gdyż wszyscy powinni mieć równe szanse. W wielu szkołach nie ma pracowni mikrokomputerowych. Natomiast uważam, że egzamin na politechnikę powinien być trudniejszy.

— Dużo osób z twojej klasy zdawało na studia?

— Przecież po liceum nie ma innego wyboru — z mojej klasy prawie wszyscy. Chociaż niektórzy podejmowali dla mnie zaskakujące decyzje: zdawali na polonistykę, bibliotekarstwo.

— Ile osób zdawało na elektronikę?

— Tylko ja jeden z klasy.

— I to z klasy o profilu matematyczno-fizycznym szkoły o takiej renomie jak XIV Liceum Ogólnokształcące im. Gottwalda! Czyżby inni uważali, że studia na elektronice są za trudne? A może szkoła nie zainteresowała ich elektroniką?

— Przyczyny są złożone. Myślę, że najpierw trzeba się zastanowić, kto trafia do szkoły. Przewa-

Rozmowa z mgr inż. Ryszardem Kottem

— asystentem w Instytucie Informatyki Politechniki Warszawskiej.

— Czym różnią się obecni studenci informatyki od tych sprzed pięciu, dziesięciu lat?

— Dziś o wiele więcej studentów pierwszego roku miało styczność z informatyką niż dziesięć lat temu — pracę w Instytucie zacząłem zaraz po uzyskaniu dyplomu w 1974 r. W Warszawie, oprócz klubów, są szkoły, które od dawna prowadzą zajęcia z informatyki np. w XIV Liceum Ogólnokształcącym im. Gottwalda, w Zespole Szkół Elektronicznych przy ul. Gen. Zajączka, czy w tradycyjnie nazywanym — Technikum Kasprzaka. Absolwenci z tych szkół mają w miarę dobre pojęcie o informatyce. Ale lepiej jest nie tylko z wiedzą teoretyczną. Zajęcia ze studentami pierwszego roku zaczynam od wywiadu i okazuje się, że coraz więcej z nich nie tylko zna pewne podstawy teoretyczne, ale ma kontakt ze sprzętem — i to coraz lepszym. Pięć lat temu w grupie studenckiej było kilka osób, które miały ograniczony dostęp do komputera. Natomiast dziś w grupie jest kilka osób, które mają własny komputer.

— Czy w związku z tym studenci rozpoczynający naukę mają sprecyzowane plany życiowe, zawodowe, naukowe związane ze studiowaniem informatyki?

— Z tym jest bardzo różnie. Już na pierwszym semestrze są studenci niemal na poziomie profesjonalnym, niektórych można porównywać z absolwentami. Oni mają dość jasny pogląd, co chcą robić w przyszłości. W dużym stopniu na decyzję o podjęciu studiów ma moda. Kandydaci, którzy się nią kierują wychodzą z założenia: informatyka to coś bardzo nowoczesnego, w co trzeba inwestować. Ale co będą robili po studiach — to się okaże.

— A więc moda utrudnia rekrutację, wybieranie najzdolniejszych?

— Tak, ale moda ma także dobre strony. Im więcej osób dobija się na nasz kierunek — tym więcej można wybrać naprawdę zdolnych. Gdy na początku lat osiemdziesiątych był dołek rekrutacyjny zauważyłem, że studenci byli słabsi.

— Skoro przyjmuje się więcej lepszych — bo wybór jest większy — to mniej studentów odpada po pierwszym semestrze?

— Ilość odpadających jest w zasadzie stała, choć np. grupa przyjęta w październiku 1985 roku wyróżniała się dobrym przygotowaniem.

— Jak ocenia pan udział szkoły w przygotowaniu kandydatów, w rozwijaniu ich zainteresowań informatyką?

— Wbrew pozorom udział szkoły jest dość istotny. Nawet jeśli uczniowie nie mają łatwego, bezpośredniego dostępu do komputera — wiadomo: sprzętu jest mało, są kłopoty z oprogramowaniem — to jednak dla nich komputer i informatyka nie są pojęciami zupełnie nowymi. Natomiast zdecydowanie lepsze przygotowanie mają osoby, które miały w szkole lub klubie łatwy dostęp do sprzętu — albo mają własny. Oni wyraźnie wybijają się już na pierwszych zajęciach. Na szczęście potem ta różnica się niweluje.

GRA O JUTRO

Dokończenie rozmowy z Norbertem Krzakiem

żnie uczniowie, którzy mają w niej starsze rodzeństwo. Duża część kieruje się panującą modą — szkoła nasza jest znana. Później są rozczarowania, gdyż w „Gottwaldzie" matematyki jest bardzo dużo. Tylko najodporniejsi nie mają dość matematyki po czterech latach.

— Ale przecież matematyka to nie wszystko. Jest jeszcze na przykład informatyka.

— Zabrzmi to zaskakująco, ale nie mieliśmy w ogóle informatyki! Na lekcjach fizyki nauczyciel trochę próbował przybliżyć nam informatykę. Ale to była tylko namiastka — komputer był wykorzystywany do ilustrowania omawianych problemów. Na przykład była lekcja o ruchu harmonicznym — wtedy nauczyciel podawał nam schemat programu, który skrupulatnie wpisywaliśmy do komputera — a potem z przyjemnością patrzyliśmy na efekty widoczne na ekranie. To wszystko. Mieliśmy w szkole pracownię mikrokomputerową, ale była ona cały czas oblegana i w niej raptem cztery ZX SPECTRUM, z których szybko dwa się spaliły a to, co ocalało, powędrowało pod klucz. Zresztą tłok w pracowni był tak wielki, że mnie zniechęcał. Ja chciałem poznać tajniki programowania, numery, które można robić z komputerami — a ludzi interesowały gry.

— Więc jak rozwijałeś swoje zainteresowania informatyką?

— W długie zimowe wieczory czytałem książki, układałem programy. Dziś najbardziej fascynuje mnie PASCAL, gdyż BASIC jest zbyt prymitywny.

— A potem nową wiedzę sprawdzałeś na komputerze?

— Niestety, nie mam komputera. I to jest mój największy problem. Praktycznie pracuję na sucho... czysto teoretycznie.

— Czy nie masz z tego powodu kompleksu? W twojej grupie na pewno będą ludzie, którzy mają własny sprzęt i solidne przygotowanie informatyczne.

— Kompleksy to za dużo powiedziane. Zdaję sobie sprawę, że na początek będę musiał więcej od nich popracować. Może w czasie studiów uda mi się wyjechać za granicę, trochę zarobić i kupić komputer, który tak kosztuje kilkaset dolarów, a u nas kilkaset tysięcy. Niestety — pod względem sprzętu, jego poziomu i produkcji jesteśmy sporo w tyle. To, co na świecie było modne dziesięć lat temu — u nas ciągle jest w cenie.

— Czy to nie zniechęca cię do studiów na politechnice?

— Z zamiarem studiowania na politechnice poszedłem do liceum. Obserwując to, co dzieje się na świecie, myślę, że nowoczesny sprzęt kiedyś dotrze także i do nas.

— Czy miałeś jakąś styczność z takim nowoczesnym sprzętem?

— Nie, w szkole — jak już mówiłem — były ZX SPECTRUM. Nie urządzano żadnych wycieczek do pracowni informatycznych. Szkoła nauczyła mnie dobrze matematyki, fizyki — i to wszystko. Swoją wiedzę informatyczną — tuż przed pójściem na studia — oceniam jako bardzo skromne minimum. Mam ogólne pojęcie — i to dość mgliste — co to jest informatyka. Uważam, że można było lepiej wykorzystać te cztery lata w ogólniaku. W takich klasach jak matematyczno-fizyczna przynajmniej raz w tygodniu przez godzinę powinna być wykładana informatyka. Tego wymagają późniejsze studia absolwentów liceum. Także wymaga tego praca późniejszych absolwentów uczelni. Ta edukacja powinna zaczynać się już od pierwszej klasy szkoły średniej. Pozwoli to później uniknąć ludziom rozczarowań, pozwoli na bardziej świadome decyzje w podejmowaniu studiów.

— A jak sobie wyobrażasz studia na kierunku informatyki?

— Przede wszystkim oczekuję, że zostanę dobrym inżynierem. Zajęcia? Koniecznie swobodny dostęp do komputera, bez tłoku, spokojnie można bez pośpiechu układać własne programy, później je sprawdzić. Uzyskać pomoc asystentów, nawet profesorów. Ważne jest zapewnienie możliwości rozwijania zainteresowań: dużo lektury z całego świata, możliwość poznania najnowszych języków programowania, najnowszego sprzętu. Chociaż wiadomo, że są kłopoty z dewizami — to przynajmniej powinna być możliwość poznawania opisów technicznych, konstrukcyjnych.

— Na jakim komputerze chciałbyś pracować?

— Myślę, że najlepszy byłby IBM PC.

— Jak wyobrażasz sobie zastosowanie wiedzy zdobytej na studiach?

— Na razie nie widzę barier w zorganizowaniu przyszłości, znalezieniu pracy. Dla mnie jest to zbyt odległa perspektywa. Poza tym wszystko teraz zależy od studiów; od tego, co uda mi się zobaczyć, jaki sprzęt poznać, jakich języków programowania nauczyć. Studia zadecydują, co będę robił jako informatyk.

Dokończenie rozmowy z mgr inż. Ryszardem Kottem

— Czy Instytut współpracuje ściślej ze szkołami?

— O ile mi wiadomo — nie ...

— Czyżby szkoły nie były zainteresowane?

— Gdyby szkoły chciały ściśle z nami współpracować, gdyby uczniowie zaczęli licznie przychodzić — to podejrzewam, że zaraz okazałoby się, iż instytut nie jest w stanie tego udźwignąć. Po pierwsze, baza jest tak skromna, że ledwie starcza na potrzeby dydaktyki i prac badawczo-rozwojowych. Po drugie: sprzęt jest przestarzały lub się sypie — tak jest eksploatowany. Po trzecie: za mało mamy kadry naukowej. W mniejszym stopniu ogranicza nas brak pomieszczeń — chociaż jest to bolączka całego wydziału. Gdyby zrodziła się świetna idea, aby raz na tydzień oddelegować dwie — trzy osoby do szkół — to obawiam się, że okazałoby się, iż nie ma kto iść.

— A akcje otwartych drzwi?

— Taką akcję można zrobić raz na rok. Przychodzą tłumy ludzi, których oprowadza się jak wycieczki po muzeum. Więcej nie jesteśmy w stanie zrobić. Niemożliwe byłoby nawet udostępnianie szkołom naszego laboratorium raz w tygodniu na kilka godzin. Przecież dostęp do sprzętu trzeba sobie rezerwować z wyprzedzeniem. Nie jesteśmy w stanie zapewnić formalnej interakcyjnej pracy wszystkim studentom — wymagamy, żeby programy były pisane na formularzach i później panie kodystki wklepują to do komputera ...

— ... na perforatory?

— Tak źle nie jest! Mamy już terminale! Jeśli program jest długi, to student kodowałby go wiele godzin. Na to nie możemy sobie pozwolić.

— Chociaż ze względów dydaktycznych byłoby lepiej, żeby pomęczył się nad klawiaturą.

— Oczywiście! Prowadziłem zajęcia za granicą, gdzie zupełnie świeżych ludzi sadzało się przy terminalach i wykład prowadzono równolegle z zajęciami praktycznymi. Tutaj nie możemy o tym marzyć!

— Czyli studenci nie mają czasu na robienie własnych programów badawczych?

— Programy badawcze — to za dużo powiedziane! Rzecz jasna, że zapaleńcy — mówię o początkujących studentach, gdyż starszych wciągamy do prac instytutu — którzy ciągle chcą robić coś samodzielnie. Jednakże często mają zupełnie błędne mniemanie o tym, co jest łatwe — co trudne, co możliwe do zrobienia — a co nie możliwe. Ale jeśli okaże się, że pomysł jest ciekawy — wówczas w ramach laboratorium wynajdujemy odrobinę czasu. Z punktu widzenia finansowania jest to niezauważalne. Nawiasem mówiąc, takich zapaleńców nie jest dużo: dwie, trzy osoby w grupie.

— Mówiliśmy, że coraz więcej studentów ma komputery. Czy jest to podstawowy wymóg dla studenta informatyki?

— Nie, z punktu widzenia dydaktyki i prac instytutu nie ma to większego znaczenia. Ale z punktu widzenia indywidualnego rozwoju — taki student ma znaczną przewagę nad kolegami. Na szczęście po wstępnym okresie, gdy się zapoznają — studenci pożyczają sobie sprzęt, wymieniają oprogramowanie. Chociaż ci, którzy nie mają własnego komputera mają gorszy start.

— A więc studenci skarżą się na zbyt trudny dostęp do komputerów?

— Jest to motyw wielu spotkań kadry dydaktycznej ze studentami, którzy są wręcz rozgoryczeni sytuacją. Każdy z nich — przychodząc na studia — miał wyobrażenia o sprzęcie, o możliwościach samodzielnej pracy. Ideałem byłoby, gdyby to, co zastanie na uczelni, przerastało owe marzenia. A już niedopuszczalne jest, aby czuł się rozczarowany.

Rozmawiali: Sławomir Polak, Roman Wojciechowski

Już jest. Nie „podobny" i nie „coś w rodzaju..., w zastępstwie". MAZOVIA 1016 — profesjonalny 16-bitowy, polski mikrokomputer odpowiadający standardowi komputera IBM PC/XT.

WSPÓLNYMI SIŁAMI

— Doszliśmy wtedy do wniosku, że skoro nikt w Polsce sam nie zrobi profesjonalnego komputera o światowym standardzie, zrobimy go wspólnie — wspomina **Andrzej Bibiński**, z-ca dyrektora ds. technicznych spółki „Mikrokomputery". — Inicjatorem powstania spółki (1 kwietnia 1985 r.) były m.in. przedsiębiorstwa: warszawska ERA i POLON, MERA-BŁONIE, POLCOLOR (Piaseczno), MERA-SYSTEM, MERAL, BIUROTECHNIKA oraz stołeczny Instytut Maszyn Matematycznych.

Dzisiaj udziałowcami spółki jest 12 przedsiębiorstw i jednostek naukowo-badawczych. Od 1985 r. pracuje Międzyzakładowy Zespół Specjalistów koordynujący, za pośrednictwem zakładowych delegatur, naukowo produkcyjny program — MAZOVIA.

— Pomysł wykrystalizował się w Instytucie Maszyn Matematycznych. Większość wyrobów na światowym rynku komputerowym jest dziełem IBM tak jak i większość programów napisana jest pod ten system — informuje inż. **Krzysztof Dzik**, kierownik zakładu w IMM. Po rozmowach dr inż. **Bronisława Piwowara**, szefa IMM, z programistą **Janem Klimowiczem**, po seminariach i sporach wybrano typ oprogramowania. Wiedzieliśmy już, że tak właśnie na żądanie programistów powstanie mikrokomputer takiej, a nie innej klasy.

Programiści zostali wysłuchani. MAZOVIA 1016 może dzięki temu pracować np. na rozpowszechnionym w świecie oprogramowaniu użytkowym do IBM PC, może również z powodzeniem strawić program w języku polskim. Na klawiaturze mikrokomputera widnieje polski alfabet, na ekranie monitora pojawiają się polskie słowa, z nich drukarka układa tekst.

Równolegle z pierwszymi dostępnymi na rynku egzemplarzami MAZOVI ma pojawić się polska dokumentacja obsługi i oprogramowania.

MŁODZI ZDOLNI

Opracowaniem oprogramowania zajęła się warszawska MERA-SYSTEM (m.in. programy funkcjonowania szpitali). Stołeczny MERAL konstruuje programy ułatwiające np. prowadzenie gospodarki materiałowej, funkcjonowanie magazynu itp. ORGMASZ opracowuje oprogramowanie niektórych stanowisk pracy w przemyśle (np. stanowisko mistrza). Z kolei Instytut Maszyn Matematycznych umożliwił komunikację z samym komputerem opracował oprogramowania systemowe i narzędziowe.

— Zintegrowany pakiet programu narzędziowego warunkuje stosowanie innych programów: np.

NIE TYL

— Zapytajcie dzisiejszych nastoletnich fanów mikrokomputerów, co to jest „Odra". Najpierw krzykną: rzeka! Dopiero później przyjdzie im do głowy, że jest to polski komputer. Ale jak wygląda, co potrafi — tego nie będzie wiedział prawie nikt.

Mgr **Piotr Zajączkowski** — kierownik **Laboratorium Informatyki** w **Instytucie Organizacji Systemów Produkcyjnych** na **Wydziale Mechaniczno-Technologicznym Politechniki Warszawskiej** nie ma wątpliwości: pogoń za szybkością operacji, fascynacja miniaturyzacją powodują nieco pobłażliwy stosunek do większych urządzeń. Dla młodych „Odra" to staroć taki, jak grająca szafa. A przecież nawet dwadzieścia małych i to najlepszych komputerów nie da tego, co jeden duży.

A „Odra" rzeczywiście jest spora. W labiryncie korytarzy na Wydziale Mechaniczno-Technologicznym odnajdujemy szkane drzwi z napisem: „Odra 1305". Za nim sala wielkości 10 m na 15 m, zajmująca tyle miejsca co cztery typowe M3 w nowym budownictwie. Wewnątrz komputer wraz z urządzeniami towarzyszącymi.

„Odra 1305" jest nie tylko duża, ale także wymagająca. A więc — musi mieć pomieszczenie klimatyzowane. Temperatura powietrza 20 stopni Celsjusza, wilgotność ok. 60 proc. Nie może być mniejsza ani większa ze względu na ładunki elektryczne, które zbierałyby się na powierzchni urządzeń. Druga sprawa — to zasilanie. Posiadacz ZX Spectrum wsadza wtyczkę do gniazdka sieciowego i ma problem z głowy. „Odra 1305" natomiast nie toleruje nawet najmniejszych odchyleń od przyjętych para-



POLSKI IBM

pozwala sporządzać arkusz rachunkowy (księgowość), wykresy graficzne — wyjaśnia **inż. Aleksander Szymerski** z MZS w IMM. Podczas tworzenia programu założyłem, że będą mogli z niego korzystać np. pracownicy służb gospodarczych czy ekonomiści w zakładzie produkcyjnym. Program pozwala również sprawdzać wyniki obliczeń przy zastosowaniu zmiennych danych (gry symulacyjne), czyli pomaga w pracy naukowcowi lub inżynierowi.

Aleksander Szymerski tworzył pakiet programu narzędziowego „Lotos A-B-C" przez siedem miesięcy. Współautorem pakietu jest **mgr Jolanta Bysiek**.

— Opracowaliśmy też podręcznik „Lotos A-B-C". Nie jest to książka do poduszki, ale ma na celu prowadzenie za rękę człowieka nie będącego informatykiem, który pracuje na MAZOVI — tłumaczy **inż. A. Szymerski**. Podręcznik pomaga m.in. poznać program narzędziowy zakodowany na dyskietce.

W 48-K pamięci ROM MAZOVI mieści się m.in. interpreter języka BASIC. Opis samego języka wraz z programowaniem grafiki typu HERKULES zawiera drugi podręcznik pt. „HBASIC — dyskietkowy interpreter BASIC". Jest to dzieło kolejnej dwójki młodych pracowników IMM: **inż. Alicji Nadrowskiej** i **inż. Wojciecha Mireckiego**. Większość członków instytutowego MZS zaledwie przed rokiem lub dwoma latami obroniła prace dyplomowe na Politechnice lub Uniwersytecie Warszawskim (elektronika, fizyka, matematyka).

Efektem pracy „młodzieżowego" tandemu inżynierów **Ireneusza Wojtkowskiego** i **Jerzego Andrzeja Szaniawskiego** jest system operacyjny PC/MS DOS. Zdaniem I. Wojtkowskiego DOS, opisany w specjalnym podręczniku, staje się tak łatwym sposobem komunikacji z komputerem, iż „młoda zdolna księgowa" jest w stanie opanować podstawowe zasady w ciągu jednego dnia. Oczywiście spółka „Mikrokomputery" oferuje szkolenie dłuższe z reguły trwające tydzień a opłata jest wliczona w cenę komputera.

RECEPTA NA NADZIENIE

Jednostka centralna MAZOVI opracowana również w IMM może zawierać m.in. pakiet procesora z 16-bitowym mikroprocesorem INTEL 8086. Od przyszłego roku będą jednak stosowane wyłącznie radzieckie mikroprocesory K 1810 WM86.

Na razie do MAZOVI dwie pamięci na dyskietkach 5.25 cala o pojemności 360 KB są sprowadzane z NRD lub krajów kapitalistycznych. Krakowska Fabryka Aparatów Pomiarowych uruchomi produkcję pamięci na dyskach elastycznych najbardziej odpowiednich do MAZOVI dopiero po 1990 r. Tyle mniej więcej musimy czekać na pamięć na dyskach twardych typu WINCHESTER 5.25 cala (10-30 MB). Dzisiaj dyski tego typu importujemy.

— Ponieważ dane w pamięci WINCHESTER mogą zostać zatarte — wyjaśnia **dyr. A. Bibiński** — będziemy oferować jako „archiwum" pamięć taśmową typu STERAMER o pojemności 60 MB. Wytwarzać ją będzie zakład MERA-MAT. Z MAZOVIĄ może współdziałać drukarka D-100/PC (80 znaków/s) będąca odmianą konstrukcyjną drukarki mozaikowej D100 wytwarzanej przez Zakłady MERA-BŁONIE. Klawiatura naszego mikrokomputera jest kompatybilna z klawiaturą wzorca. Korzystając z 84 klawiszy można pisać z szybkością 25 znaków na sekundę w języku polskim, rosyjskim lub angielskim.

W październiku br. z 500 seryjnych „polskich IBM-ów" 300 zostanie sprzedanych krajowym użytkownikom. Reszta trafi do udziałowców spółki „Mikrokomputery", programistów oraz do siedmiu przedsiębiorstw serwisowych BIUROTECHNIKI. Za rok spółka ma zamiar zmontować ok. 3 tysięcy mikrokomputerów.

MAZOVIA wraz z kompletem urządzeń zewnętrznych będzie kosztować ok. 3 miliony złotych. Od 1987 r. ilość twardej waluty tkwiącej w MAZOVI nie powinna przekraczać 10-15 proc. jej wartości.

Wojciech Gładykowski

...KO MAŁE

metrów napięcia prądu. Dlatego zainstalowano w pobliżu przetwornicę prądu, żeby nie było zakłóceń. Polega to na tym, że prąd z sieci zasila silnik, który napędza generator i... jest własny prąd, do którego można już mieć zaufanie.

Ale to nie wszystko, jeśli chodzi o wymagania „Odry 1305". Jeszcze jedna ciekawostka — komputer nie znosi kurzu, ze względu na dyski i taśmy magnetyczne. Dlatego w pomieszczeniu panuje cały czas nadciśnienie — czyste powietrze jest wtłaczane przez specjalne filtry. Gdy ktoś wchodzi przez drzwi, czyste powietrze wywiewa kurz z korytarza.

Wydział Mechaniczno-Technologiczny ma kilka „dużych maszyn": dwa **Riady 32"** i trzy **„Odry 1305"**. Niestety, „Odry" — delikatnie określając — nie są urządzeniami najnowszymi w swojej klasie. Pochodzą z początku lat siedemdziesiątych, a w pracowni zostały zainstalowane w 1976 roku. W sensie logicznym są to maszyny identyczne jak **ICL 1900** i przez pewien czas oprogramowanie i wyposażenie dodatkowe sprowadzano z Anglii. Mimo, że **„Odra 1305"** nie jest najmłodsza i przez to nowoczesna, zdaniem **Piotra Zajączkowskiego** jest to w uczelni w zasadzie jedyny komputer, na którym można zbudować sensowny system wielodostępny. Do „Odry 1305" podłączonych jest pięćdziesiąt terminali i to na trzech wydziałach: **Mechaniki Precyzyjnej**, **Mechaniczno-Technologicznym** i wydziale **Samochody i Ciągniki**. Niestety, poza tzw. południowym terenem Politechniki, **„Odra 1305"** nie ma teletransmisji. Ale nie jest to już zależne od uczelni.

Co takie duże urządzenie potrafi? Przede wszystkim ma nieco inną strukturę niż mały komputer, który może składać się z kilku pakietów a na każdym z nich może być po kilka mikroprocesorów. „Odra 1305" to zestaw bardzo wielu podzespołów. Pamięć mierzy się zupełnie innymi miarkami: w systemie słów. Komputer dysponuje pamięcią 512 kilosłów. Do dyspozycji użytkownika jest także arytmometr z układem operacji stało i zmiennoprzecinkowych w zakresie jedenastu cyfr dziesiętnych.

Obsługa pamięci jest dwojaka: na dyskach i taśmach magnetycznych. Cztery sterowniki obsługują osiem dysków o pojemności 8 Megabajtów każdy. Średni dostęp do informacji — 110 milisekund. Dla porównania **Winchester** 20MB" do **IBM** ma czas dostępu 80 milisekund, w modelach bardziej zaawansowanych — 20 milisekund. **„Odra 1305"** pracuje w systemie **„George 3"**.

Gdy na dyskach robi się tłok, a za przechowywanie informacji użytkownicy (zakłady, instytucje zlecające opracowanie programów) przecież płacą. Przenosi się programy na taśmy magnetyczne. Te przechowuje się w sąsiednim magazynie. Komputer pracuje wykorzystując taśmę niezbyt oszczędnie: 800 bitów na cal. Tu znowu nietypowy system miar, gdyż taśmy BASF mierzone są w calach i mają... 2400 stóp długości, a także maksymalną możliwość zapisywania 6250 bitów na celu. Odszukanie informacji zapisanej na taśmie nie przekracza pięciu minut. Może wydać się to długo, ale w przypadku programów rzadko używanych, wręcz archiwalnych, jest to czas zupełnie przyzwoity.

Z urządzeń towarzyszących warto jeszcze wymienić **„Multiplexer"** pozwalający podłączyć do **„Odry 1305"** 63 terminale, w tym także monitory do niej fabrycznie nieprzystosowane. Ze względów dydaktycznych używa się także ...czytnika kart „MERA 7946". Po prostu studenci nie potrafią pisać na maszynie i korzystanie z komputerowej klawiatury sprawia im zbyt dużo kłopotów. Pewien respekt budzą natomiast drukarki wierszowe o wydajności 1100 wierszy na minutę — w każdym wierszu 160 znaków. Do rysowania wykresów używa się plotera — model także niezbyt nowoczesny, ale za to niezawodny, o sprawności 300 kroków na sekundę, a każdy krok to 0,1 mm. Najczęściej wykorzystywany jest on do wykresów optycznych.



To wszystko nie zmieści się na biurku.

„Odra 1305" nie jest komputerem nowoczesnym, a jednak nie można sobie wyobrazić pracy Instytutu bez urządzenia tej wielkości. Na całym świecie używa się w uczelniach i placówkach naukowo-badawczych superkomputerów typu **VAX**, które mogą pracować w sieci, ze 120 terminalami. W wypadku takich urządzeń nikt nie pyta o pamięć operacyjną — jest tak duża, że po prostu nie ma praktycznych ograniczeń.

„Odra 1305" ma już wyznaczoną granicę swojej egzystencji w uczelni — jest nią rok 1990. Przynajmniej tak się planuje. Po tym okresie, Laboratorium Informatyki ma przejść na inny system, bardziej nowoczesny. Czy będzie to **IBM**, **WANG**, **CDC**, **VAX**...? W każdym razie znów będzie to komputer duży.

Sławomir Polak
Roman Wojciechowski

Do prezentowanych dotychczas na łamach „Bajtka" języków programowania: BASIC, LOGO i MicroPROLOG dołączamy następny — PASCAL. Zachęcamy do zapoznania się z nim, bowiem w porównaniu z wyżej wymienionymi językami otwiera on przed użytkownikiem mikrokomputera całą gamę nowych możliwości.

Język **PASCAL** został opracowany w roku 1971 przez znanego szwajcarskiego informatyka **Niklausa Wirtha.** Oryginalny raport języka wydany wówczas zawiera dość ścisły jego opis, niemniej spotkać można dziś sporo różnych wersji **PASCAL-a** — do dziś nie został ustalony jego obowiązujący standard. Wiele uzupełnień i innowacji spowodowanych zostało przez konieczność dostosowania operacji wejścia/wyjścia do wprowadzonego w międzyczasie nowego rodzaju sprzętu monitora ekranowego, klawiatury, pojedynczej stacji dysków elastycznych.

Z tego powodu zajmiemy się tutaj programami zgodnymi z pierwotną wersją języka, zaznaczając jedynie miejsca, w których mogą wystąpić różnice pomiędzy wersjami nowszymi — bo z takimi spotyka się użytkownik mikrokomputera.

OPROGRAMOWANIE

PASCAL jest obecnie dostępny dla wszystkich popularnych typów mikrokomputerów. Wymienić tu można np. następujące jego kompilatory: dla **ZX Spectrum 48 k — Hisoft Pascal** (na kasecie); dla **Commodore 64** — sporo różnych kompilatorów wymagających stacji dysków, np. **Profi-Pascal, Oxford Pascal;** dla **CPC-464 — Hisoft Pascal** (na kasecie).

Osobno wymienić należy bardzo dobry **Turbo PASCAL** firmy **Boreland Inc.**, dostępny pod systemem **CP/M 2.2** i **CP/M Plus** (a zatem dla wszystkich Amstradów ze stacją dysków oraz Commodore 128) oraz — w swojej pełnej i najdoskonalszej wersji — pod systemem **MS DOS** (dla IBM PC).

Wszystkie omawiane tu przykłady programów zostały napisane w **Turbo PASCAL-u** i przetestowane na komputerze **CPC 6128.** Ich tematyka i treść została jednak tak dobrana, aby mogły być bez przeszkód uruchomione na innych typach komputerów.

LITERATURA

Szczupłość miejsca nie pozwoli przedstawić nam w tym krótkim cyklu wszystkich możliwości i zalet **PASCAL-a.** Na szczęście możemy w tym miejscu polecić ogólnodostępną lekturę. Jest nią książka **M. Iglewski, J. Madey, S. Matwin „Pascal", wyd.III, WNT 1984.** Pozycja ta zawiera kompletny opis języka wzorcowego, wzbogacony o liczne przykłady programów. Dla czytelników nieprzygotowanych może ona być lekturą trudną, niemniej wypada polecić ją jako „przewodnik".

PISANIE PROGRAMU

W przypadku **PASCAL-a** mamy do czynienia z programem kompilowanym. Zaletą tego rozwiązania jest duża szybkość wykonywania gotowego programu — kilka do kilkadziesiąt razy szybciej niż w przypadku interpretacji. W przypadku komputera bez stacji dysków ma to również dużą wadę: w ograniczonej objętości pamięci trzeba zmieścić kompilator, tekst źródłowy programu w **PASCAL-u** oraz jego kod wynikowy. Stąd ograniczenie objętości programów, nieraz tylko do kilku K.

Większość dostępnych na mikrokomputerach kompilatorów obsługiwanych jest przy pomocy „menu". Uruchomienie programu wygląda więc w sposób następujący:

1. Uruchomienie edytora tekstu i napisanie programu. Spotkać się można z różnymi edytorami, niektóre z nich wymagają numerowania linii programu. Należy pamiętać, że numery linii nie mają żadnego znaczenia dla samego programu (odwrotnie, niż w **BASIC-u**), lecz pełnią wyłącznie funkcję pomocniczą.
2. Po zakończeniu wpisywania programu powracamy do „głównego menu" i uruchamiamy jego kompilację — najczęściej poprzez wciśnięcie klawisza **C** (od „**C**ompile").
3. W przypadku pomyślnego zakończenia kompilacji uruchamiamy program.
4. W przypadku napotkania przez kompilator błędu w tekście programu najczęściej wskazywane jest miejsce jego wystąpienia oraz określany typ błędu. Stanowi to podstawę do dokonania poprawki (ponownie używamy edytora), po której można ponownie przystąpić do kompilacji.

Taki sposób pracy wymaga uprzedniego zapoznania się z obsługą samego kompilatora. W szczególności należy posiąść umiejętność:

- nagrania swojego programu na taśmie lub dysku,
- korzystania z edytora tekstu,
- skompilowania i uruchomienia programu.

PIERWSZE PROGRAMY

Po tym długim, lecz potrzebnym wstępie przystąpmy od razu do pisania programu. Będzie on bardzo prosty — wypisze na ekranie kilka słów przywitania.

Każdy program pascalowy rozpoczyna się od nagłówka.

```
program DzienDobry;
```

Nazwę programu można wybrać dowolnie — musi to być jedynie ciąg liter i cyfr, rozpoczynający się od litery. Pamiętać należy o średniku kończącym nagłówek.

Właściwy program rozpoczynamy dodatkowo słowem

```
begin
```

.. i następnie piszemy ciąg instrukcji, z których nasz program jest złożony. W tym przypadku poprzestaniemy na jednej — poleceniu wypisania na ekranie kilku słów

```
writeln (' Dzien dobry. Milo Cie poznac. ');
```

Koniec programu oznaczamy słowem „end", po którym następuje kropka (a nie średnik)

```
end.
```

Cały nasz program przedstawia się teraz następująco:

```
program DzienDobry;

begin
  writeln (' Dzien dobry. Milo Cie poznac. ');
end.
```



Kompilujemy teraz całość i po wykonaniu programu otrzymujemy na ekranie napis

```
Dzien dobry. Milo Cie Poznac.
```

Użyta przez nas w programie procedura **WRITELN** (skrót od **WRITE LiNe**) powoduje, że wypisany zostaje podany jako parametr ciąg znaków, po czym kursor zostaje przeniesiony do początku nowego wiersza. W przypadku, gdy chcemy owej zmiany wiersza uniknąć, należy zamiast **WRITELN** użyć **WRITE.** Z kolei podanie **WRITELN** bez żadnego parametru powoduje tylko przejście do nowego wiersza.

Wiedząc to, możemy nasz program nieco zmodyfikować.

```
program DzienDobry;

begin
  write (' --- ');
  writeln (' Dzien Dobry. ');
  writeln;
  writeln (' Milo mi Ciebie poznac. ')
end.
```

Pisząc dłuższą wersję programu należy pamiętać, że w Pascalu każda instrukcja musi być zakończona średnikiem. Wyjątkiem od tej zasady jest sytuacja, gdy bezpośrednio po instrukcji występuje słowo **END** (przekonamy się później, że może ono wystąpić nie tylko na końcu programu). Wówczas średnik nie jest potrzebny, ale i nie przeszkadza.

Na ekranie otrzymujemy teraz zmieniony wydruk.

```
---  Dzien Dobry.


Milo mi Ciebie poznac.
```

W naszym pierwszym programie użyliśmy — za wyjątkiem parametrów procedur **WRITE** i **WRITELN** — wyłącznie słów, używanych przez język standardowo. Nie wprowadziliśmy żadnych własnych pojęć, jak np. nazwy zmiennych czy własne procedury.

W **PASCAL-u** istnieje i jest rygorystycznie przestrzegana zasada tzw. predefinicji pojęć. Znaczy to, że każde występujące w programie pojęcie, nie należące do języka (a zatem np. nazwa zmiennej, procedury czy funkcji), musi być uprzednio zdefiniowane. Z tego powodu w programie pascalowym można wyróżnić dwie części: **deklaracyjną,** zawierającą owe definicje, oraz **operacyjną,** stanowiącą „program właściwy", zawarty między słowami **BEGIN...END.**

Dotychczas mieliśmy do czynienia z pustą częścią deklaracyjną.

W następnym przykładzie wprowadzimy kilka zmiennych. Napiszemy mianowicie program, który dla podanej liczby rzeczywistej oblicza i wypisuje pole koła o tym promieniu.

Przed podaniem treści programu jeszcze jedna informacja. Ciąg znaków zawarty pomię-

dzy nawiasami klamrowymi stanowi komentarz i jest ignorowany przez kompilator. Komentarze wolno umieszczać w każdym miejscu programu, w którym występuje spacja. Można je (choć nie warto!) pominąć przy przepisywaniu programu.

```
program PoleKola;

{ Program obliczajacy pole
  kola o zadanym promieniu }

const Pi = 3.1415926;

{ Definujemy stala Pi }

var Promien, Pole : real;

{ Deklarujemy uzywane w programie zmienne.
  Sa one typu rzeczywistego - real }

{ Teraz czesc operacyjna programu }

begin
  { Wczytujemy najpierw wartosc promienia }
  write (' Jaki jest promien naszego kola? ');
  read (Promien);
  writeln;
  { Obliczamy teraz pole kola i wypisujemy wynik }
  Pole := Pi * Promien * Promien;
  writeln (' Pole kola o promieniu ');
  writeln (Promien);
  writeln (' wynosi: ', Pole)
end.
```

Widzimy, że część deklaracyjna programu podzielona jest na dalsze moduły: część definiującą stałe, poprzedzoną nagłówkiem **CONST,** oraz część deklarującą zmienne, następującą po nagłówku **VAR** (skrót od **VAR**iables). Modułów takich może być w części deklaracyjnej więcej — omówimy je nieco dalej. Pamiętać trzeba, że każda zadeklarowana zmienna musi mieć przypisany określony typ. W naszym przypadku wszystkie używane (i zadeklarowane) w programie zmienne są typu rzeczywistego — **REAL.** Nazwy zmiennych oraz ich typ rozdzielamy dwukropkiem, tak, jak to jest uwidocznione w programie. W definicji stałej w analogicznym miejscu występuje znak równości.

W części operacyjnej mamy również kilka nowych rzeczy. Procedura **READ (Promien)** powoduje, że z klawiatury wczytana zostaje podana przez użytkownika liczba, a następnie jej wartość podstawiona jako wartość zmiennej **Promien**. Nie następuje przy tym przeniesienie kursora do nowego wiersza — stąd użycie **WRITELN** w następnej linii programu.

Specyficznego symbolu := używamy w **PASCAL-u** w znaczeniu „staje się" lub „przyjmuje wartość". Instrukcję

Alfa := 0;

odczytamy zatem jako „Alfa przyjmuje wartość 0". Symbol ten, zwany symbolem przypisania, niewiele ma wspólnego ze znakiem równości i nie należy go z nim mylić. Ten ostatni jest w Pascalu używany do porównywania różnych wielkości.

Wynikiem wykonania **WRITELN (Promien)** jest wydruk aktualnej wartości zmiennej **Promien.** Linię niżej widzimy przykład łączenia w jednej linii wydruku tekstu i wartości zmiennej Pole.

Po skompilowaniu programu uruchamiamy go, otrzymując mniej więcej taki efekt:

```
Jaki jest promien naszego kola? 100
Pole kola o promieniu
 1.0000000000E+02
wynosi:   3.1415926000E+04
```

Jako ćwiczenie i pomoc w utrwaleniu dotychczas zdobytych wiadomości posłuży nam program następny, obliczający pierwiastki równania kwadratowego

$$a * x^2 + b * x + c = 0$$

zgodnie z wzorem

Delta := $b^2 - 4ac$

jeśli **Delta** >=**0,** to

$$x_1 := \frac{-b + \sqrt{Delta}}{2a} \qquad x_2 := \frac{-b - \sqrt{Delta}}{2a}$$

Oto nasz program:

```
program Pierwiastki;

var a, b, c : real;
      Delta : real;
      x1, x2 : real;

begin
  { Najpierw wczytujemy wspolczynniki rownania }
  writeln (' Podaj wspolczynniki rownania - a, b, c: ');
  read (a, b, c);
  writeln;

  { Rozpatrujemy przypadek, gdy a = 0 }
  if a = 0 then
    writeln (' Jest to rownanie liniowe. Rozwiaz je sam! ')
  else          { Przypadek przeciwny: a <> 0 }
    begin
      writeln (' Rozwiazanie rownania kwadratowego postaci: ');
      writeln (a, ' * x^2 + ', b, ' * x + ', c, ' = 0 ');
      writeln;
      Delta := b * b - 4 * a * c;
      if Delta < 0 then
        writeln (' nie istnieje w zbiorze liczb rzeczywistych. ')
      else
        begin
          x1 := (-b + sqrt (Delta)) / (2 * a);
          x2 := (-b - sqrt (Delta)) / (2 * a);
          writeln (' stanowia dwa pierwiastki rzeczywiste ');
          writeln (' X1 = ', x1);
          writeln (' X2 = ', x2)
        end
    end
end.
```

W programie tym pojawia się instrukcja warunkowa postaci

IF ‹warunek› **THEN** ‹instrukcja› **ELSE** ‹instrukcja›;

którą odczytujemy: „Jeżeli ... to ..., w przeciwnym wypadku ...". Instrukcja warunkowa może również wystąpić w postaci:

IF ‹warunek› **THEN** ‹instrukcja›;

Obie postacie tej instrukcji muszą być rozróżnialne przez kompilator. Z tego powodu bezpośrednio przed słowem ELSE **nie może** występować średnik — proszę spojrzeć do tekstu programu.

W obu przypadkach użycia **IF** wystąpiła potrzeba wykonania „w przeciwnym wypadku" więcej niż jednej instrukcji. W takim wypadku można zgrupować je poprzez ujęcie ciągu instrukcji pomiędzy słowa **BEGIN**...**END.** Tworzymy wówczas tzw. instrukcję złożoną. Instrukcja złożona może oczywiście w swoim wnętrzu zawierać inne instrukcje złożone. Instrukcję złożoną kończymy również średnikiem — ze wszystkimi wymienionymi poprzednio zastrzeżeniami.

Warto zauważyć, że przy tym sposobie zapisu algorytmu właściwie cała część operacyjna programu jest jedną instrukcją złożoną (zakończoną dla odmiany kropką, o czym już wspominaliśmy).

Na koniec zajmijmy się krótko jeszcze jedną kwestią formalną. Zwróćmy uwagę, że wszystkie zaprezentowane tutaj programy zostały w specyficzny sposób zaformatowane — np. poprzez specjalne ułożenie akapitów w tekście programu. Nie jest to uwarunkowane składnią **PASCAL-a** — z punktu widzenia kompilatora jest obojętne, czy napiszemy program tak, jak to zostało uwidocznione, czy np. tak:

```
program Pierwiastki;var a, b, c : real;Delta :
real;x1, x2 : real;                    begin
writeln (' Podaj wspolczynniki rownania - a, b, c: ');
read (a, b, c);writeln;
if a = 0 then writeln (' Jest to rownanie liniowe. Rozwiaz je sam! ')
else begin   writeln (' Rozwiazanie rownania kwadratowego postaci: ');
writeln (a,' * x^2 + ',b,' * x + ',c,' = 0 ');
writeln;Delta:=b*b-4*a*c;
if Delta < 0 then writeln (' nie istnieje w zbiorze liczb rzeczywistych. ')
else begin x1 := (-b + sqrt (Delta)) / (2 * a);
x2 := (-b - sqrt (Delta)) / (2 * a);writeln
(' stanowia dwa pierwiastki rzeczywiste ');writeln (' X1 = ', x1);
writeln (' X2 = ', x2) end end end.
```

Jest to chyba jasne, że zaproponowana forma zapisu programu jest znacznie bardziej czytelna — wyróżnia ona wszystkie części programu i czyni go bardziej przejrzystym.

Za miesiąc zajmiemy się budowaniem programów bardziej skomplikowanych.

Marek Wyrwidąb

WSTĘP DO PROGRAMOWANIA W JĘZYKU PASCAL

cz. I

PROGRAM KATALOG

Ciekawym narzędziem pomocnym przy tworzeniu spisu własnej biblioteki programów jest zaprezentowany tu program zatytułowany „KATALOG". Program napisany jest dla mikrokomputera ZX SPECTRUM głównie w języku BASIC. Niewielkie fragmenty (aczkolwiek bardzo ważne z punktu widzenia poprawności działania) są napisane w kodzie maszynowym umieszczonym w instrukcji DATA.

Przy pomocy programu „KATALOG" możemy prosto i bez wysiłku utworzyć wykaz posiadanych programów łącznie z określeniem typu programu (np. BASIC, kod maszynowy, dane), długości i linii od której program startuje. Odczytane z taśmy magnetofonowej nagłówki programów uzupełniamy numerami licznika magnetofonowego, które pozwolą




```
  1 REM ***********
  2 REM * KATALOG *
  4 REM *  1986   *
  5 REM ***********
  6
 10 CLEAR 54999
 20 LET B$=" ": LET P$=" ": DIM
 a$(100,25)
 21 LET file=0
 40 RESTORE
 50 FOR a=55000 TO 55009: READ
b: POKE a,b: NEXT a
 60 DATA 175,55,221,33,16,127,2
05,86,5,201
 61 LET b=32528: DEF FN a(x)=PE
EK (b+x)+256*PEEK (b+x+1)
 62 RESTORE 64
 63 FOR a=60899 TO 60971: READ
c: POKE a,c: NEXT a
 64 DATA 205,124,0,59,59,225,1,
15,0,9,235,42,61,92,115,35,114,2
01,118,205,142,2,123,254,255,32,
248,58,58,92
 65 DATA 254,12,40,10,254,16,40
,6,254,20,40,2,24,25,60,50,129,9
2,253,54,0,255,33,23,37,34,66,92
,33,0,0,34,68
 66 DATA 92,59,59,195,125,27,19
5,3,19,0
 74 GO TO 9495
 75 REM CZYTANIE TASMY
 76 LET B$="T": INPUT "CZY WYDR
UK NA DRUKARCE T/N ";B$
 77 INPUT " ": PRINT #0;"OD NOW
A CZY DALSZY CIAG (N/D)?": PAUSE
 0: LET P$=INKEY$
 79 IF P$(1)="D" THEN  PAPER 0:
 INK 7: CLS : GO TO 100
 80 CLS : POKE 23658,8: INPUT "
Tasma No: ";tas,"Strona A/B "; L
INE s$: PAPER 0: INK 7: CLS
 81 DIM a$(100,25)
 82 LET file=0
 90 LET a$(1)(1 TO 8)="TASMA No
": LET A$(1)(10 TO 12)=STR$ tas:
 LET A$(1)(14 TO 19)="STRONA": L
ET A$(1,21)=S$
 95 IF B$="T" THEN  LPRINT ;TAB
 6;A$(1)'"*********************
**********"'"LICZ. NAZWA     TYP
   DLUG. START"'"================
================"
100 IF file>0 THEN  GO SUB 370:
 GO SUB 0390
110 GO SUB 0360
120 LET file=file+1
125 LET A$(FILE+1)(1 TO 3)="___
"
130 INPUT " ": PRINT #1; FLASH
1;"  PROSZE O DANE Z MAGNETOFONU
   ": RANDOMIZE USR 55000
140 LET c=PEEK b
150 IF C>3 THEN  LET A$(FILE+1)
(4 TO )="          *          ":
 GO TO 220
160 FOR i=1 TO 10: IF (PEEK (i+
b)>=32 AND PEEK (i+b)<=164) THEN
  LET a$(file+1,i+3)=CHR$ PEEK (
i+b)
170 NEXT i
180 IF c=0 THEN  GO SUB 0240
190 IF c=1 THEN  GO SUB 0280
200 IF c=2 THEN  GO SUB 0290
210 IF c=3 THEN  GO SUB 0320
215 POKE B,255: RANDOMIZE USR 5
5000
220 POKE b,255
230 GO TO 100
240 LET a$(file+1)(14)="P"
250 IF FN a(13)<=9999 OR c<>0 T
HEN  LET a$(file+1)(20 TO 24)=ST
R$ FN a(13)
260 LET a$(file+1)(15 TO 19)=ST
R$ FN a(11)
270 RETURN
280 LET a$(file+1,14)="A": GO T
O 0300
290 LET a$(file+1,14)="$"
300 LET a$(file+1)(20 TO 24)=CH
R$ (64+32*(PEEK (b+14)/32-INT (P
EEK (b+14)/32))+("$" AND (a$(fi
le+1,14)="$"))
310 GO TO 0260
320 LET a$(file+1,14)="B"
330 GO TO 0250
340 IF c>3 THEN  GO TO 100
360 PRINT AT 0,0;TAB 6;A$(1)'"*
*******************************"
'"LICZ. NAZWA     TYP  DLUG. STA
RT"'"===========================
====="
365 RETURN
370 POKE 23692,25: PRINT AT (fi
le-19)*(file<=19)+22,0;: IF file
>=20 THEN  PRINT " "
380 RETURN
390 PRINT a$(file+1)(1 TO 3);"
  ";a$(file+1)(4 TO 13);" ";a$(f
ile+1)(14);"   ";a$(file+1)(15 T
O 19);" ";a$(file+1)(20 TO 24)
395 IF B$="T" THEN  LPRINT a$(f
ile+1)(1 TO 3);"   ";a$(file+1)(
4 TO 13);" ";a$(file+1)(14);"
";a$(file+1)(15 TO 19);" ";a$(fi
le+1)(20 TO 24)
396 RETURN
400 REM WYDRUK NAZW NA DRUKARCE
405 INPUT "EKRAN CZY DRUKARKA ?
 E/D ";B$
406 IF B$="E" THEN  CLS : OPEN
#3,"S": GO TO 410
410 LPRINT ;TAB 6;A$(1)(1 TO 22
)'"*****************************
***": LPRINT "LICZ. NAZWA     TY
P  DLUG. START "'"==============
=================="
420 FOR i=1 TO file
430 LPRINT a$(i+1)(1 TO 3);"
";a$(i+1)(4 TO 13);" ";a$(i+1)(1
4);"   ";a$(i+1)(15 TO 19);" ";a
$(i+1)(20 TO 24)
440 NEXT i
445 IF B$(1)="E" THEN  PAUSE 0
450 GO TO 9495
500 REM ZAPIS NAGLOWKOW
508 LET A$(1)(23 TO 25)=STR$ FI
LE
515 SAVE ("TASMA"+A$(1)(10 TO 1
1)+A$(1)(21)) DATA A$(): GO TO 5
20
520 INPUT " ": PRINT #0;"CZY WE
RYFIKOWAC (T/N)?": PAUSE 0: LET
P$=INKEY$
525 IF P$="N" THEN  GO TO 9495
540 VERIFY ("TASMA"+A$(1)(10 TO
 11)+A$(1)(21)) DATA A$()
550 GO TO 9495
600 REM ZALADOWANIE ZBIORU
610 POKE 23658,8: INPUT "Tasma
No: ";tas,"Strona A/B "; LINE s$
615 DIM A$(200,25)
620 DIM P$(2): LET P$=STR$ TAS
632 LOAD ("TASMA"+P$+S$) DATA A
$()
635 LET FILE=VAL A$(1)(23 TO 25
)
640 GO TO 9495
700 REM WPISYWANIE STANOW LICZN
IKA
710 CLS : FOR I=2 TO FILE+1
730 INPUT (a$(I)(1 TO 3);"    ";
a$(I)(4 TO 13);" ";a$(I)(14);"
 ";a$(I)(15 TO 19);" ";a$(I)(20
TO 24));AT 1,0;L
732 IF L=0 THEN  GO TO 741
740 LET A$(I)(1 TO 3)=STR$ L
741 POKE 23692,255
745 PRINT a$(I)(1 TO 3);"    ";a
$(I)(4 TO 13);" ";a$(I)(14);"
```

```
";a$(I)(15 TO 19);" ";a$(I)(20 T
O 24)
 750 NEXT I
 760 GO TO 9495
 800 REM KASOWANIE CZESCI ZBIORU
 810 CLS : INPUT "OD KTOREGO ZBI
ORU WYKASOWAC? ";A
 820 INPUT (A$(A+1));" T/N";B$
 830 IF B$(1)="T" THEN  FOR I=A+
1 TO FILE+1: LET A$(I)="
                        ": NEXT I: LET F
ILE=A-1: GO TO 9495
 840 GO TO 800
9495 REM MENU
9496 CLOSE #3: CLOSE #2
9499 RANDOMIZE USR 60899
9500 BORDER 4: PAPER 4: INK 0: C
LS : PRINT TAB 13;"KATALOG"''''T
AB 4;"WYBIERZ FUNKCJE PROGRAMU "
9502 PRINT " ____________________
_________"
9510 PRINT ''" 1 CZYTANIE TASMY"
''" 2 WCZYTANIE NAGLOWKOW Z TASM
Y"''" 3 ZAPIS NAGLOWKOW"''" 4 WY
DRUK NAGLOWKOW"''" 5 WPISYWANIE
LICZNIKA"''" 6 KASOWANIE CZESCI
NAGLOWKOW"''" 7 POWROT DO BASIC'
A"
9515 PRINT #0;" 1   2   3   4
 5   6   7 "
9520 PAUSE 0: POKE 23658,8: LET
B$=INKEY$
9525 IF B$="1" THEN  CLS : GO TO
 76
9530 IF B$="2" THEN  GO TO 600
9540 IF B$="3" THEN  GO TO 500
9550 IF B$="4" THEN  GO TO 400
9560 IF B$="5" THEN  GO TO 700
9570 IF B$="6" THEN  GO TO 800
9580 IF B$="7" THEN  STOP
9600 GO TO 9520
9999 SAVE "KATALOG" LINE 10: STO
P
```

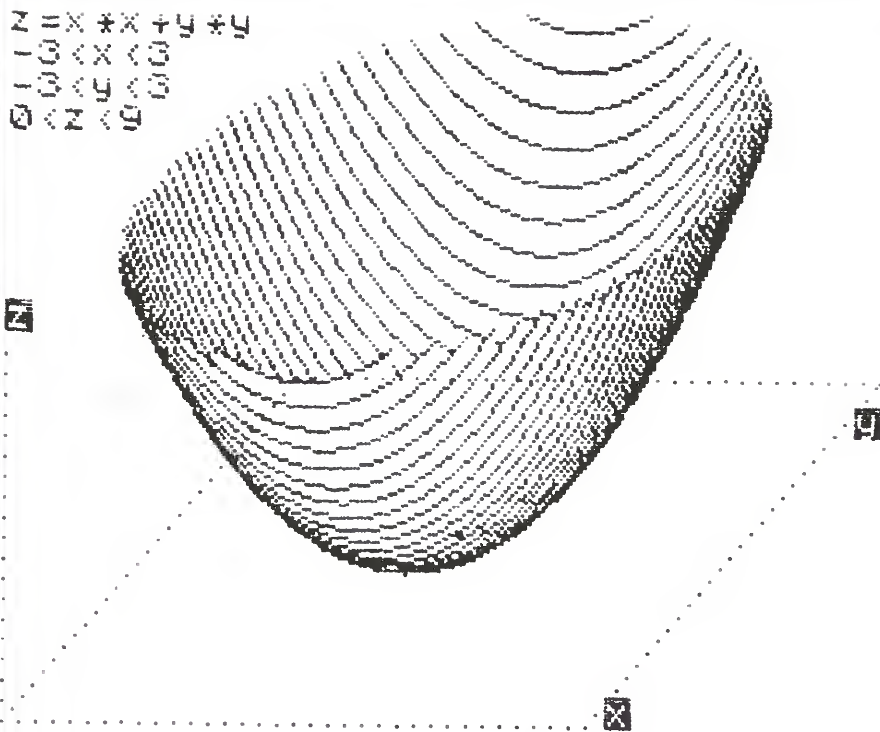
nam na szybkie odnalezienie szukanego tytułu. Wybierając odpowiednie funkcje programu możemy dokonać weryfikacji tytułów na ekranie lub wydrukować cały spis na drukarce. Program korzysta z instrukcji **LPRINT** tak jak dla drukarki **ZX PRINTER** lub **SEIKOSHA GP 50**. Mając do dyspozycji inną drukarkę należy dołączyć do programu „KATALOG" program jej obsługi, umożliwiający korzystanie z tej instrukcji. Należy przy tym pamiętać o obszarach pamięci mikrokomputera zajętych przez kod maszynowy.

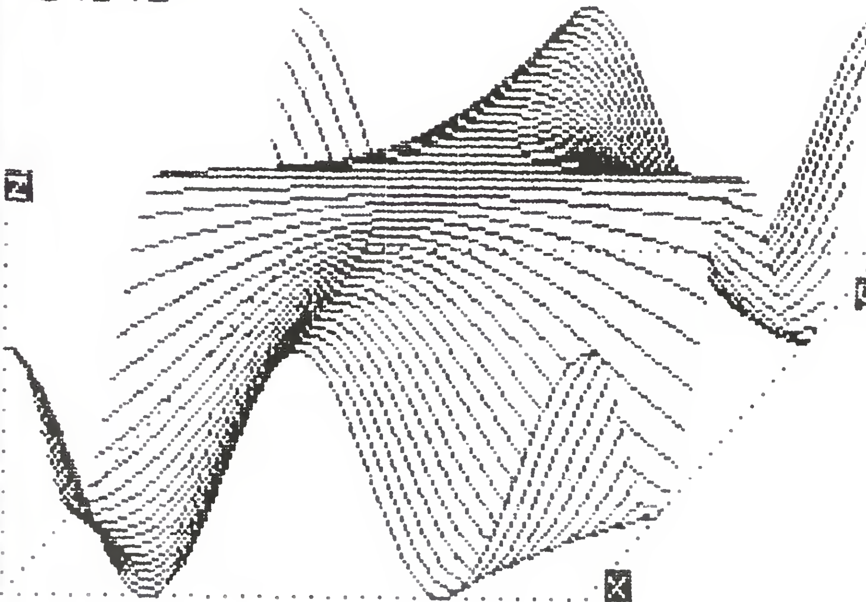
Program na taśmie możemy zapisać rozkazem **GO TO 9999**, program tak zapisany będzie uruchamiał się automatycznie po wczytaniu do mikrokomputera.

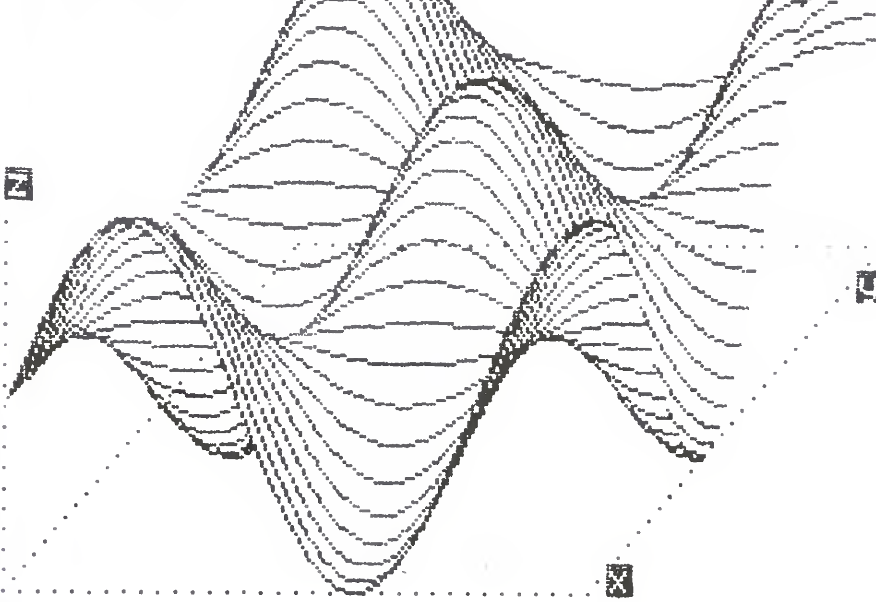
Program daje nam do dyspozycji 7 funkcji wybieranych klawiszami od 1 do 7. Zaczynamy od wybrania funkcji 1. Następnie do magnetofonu wkładamy kasetę, której spis zawartości chcemy skatalogować i uruchamiamy magnetofon (przy połączeniach przewodów tak jak do odtwarzania programów). Nagłówki napotkanych programów zostaną odczytane, określony zostanie ich typ, długość i start. Całość informacji widoczna będzie na ekranie. Następnie wybierając funkcję 5 możemy wpisać stany liczników. Po sprawdzeniu całości na ekranie funkcją 4, gotowy spis możemy wydrukować na drukarce.

Możliwe jest też zachowanie całego spisu kasety w postaci bloku danych na taśmie, które to dane mogą być wykorzystane przy drukowaniu uaktualnianego spisu.

Konrad Fedyna
Zygmunt Wereszczyński
Andrzej Ciepliński









Ten program zrealizowany na ZX Spectrum, umożliwia uzyskanie pseudo-przestrzennego wykresu funkcji dówch zmiennych **z=f(x,y)** jest napisany wyłącznie w języku BASIC i po niewielkich przeróbkach może działać także na innym mikrokomputerze posiadającym wysokorozdzielczą grafikę.

Obsługa programu jest bardzo prosta: należy tylko wprowadzić jako dane:

1) wzór funkcji używając jako argumentów x i y. Np. aby otrzymać funkcję **z=sin(xy)** trzeba wprowadzć **SIN ABS (x×y).**
2) Zakresy zmiennych x,y,z (jeżeli funkcja wykracza za zakres zmiennej zostaje obcięta).
3) Dziedzinę funkcji. Np. jeżeli funkcja jest nieokreślona w punkcie (0,0), należy wprowadzić:
   x <> 0 AND y <> 0 (AND jest tutaj słowem kluczowym). Jeżeli funkcja jest określona na całej płaszczyźnie wystarczy wcisnąć ENTER. Możliwe jest również wprowadzanie dziedziny w postaci uwikłanej np.
   2×x+3×y+5<>0

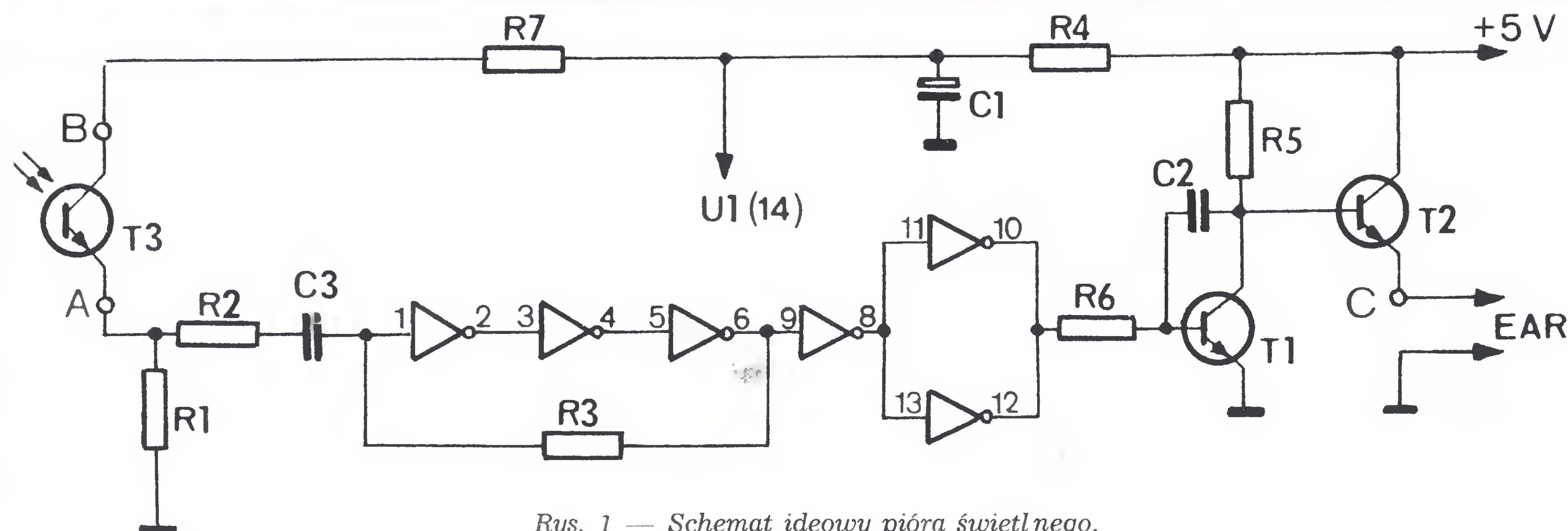
Szymon Słupik

```
  10 BORDER 0: PAPER 0: INK 9: C
LS
  20 DIM z(255): DIM y(255)
  30 FOR n=1 TO 255: LET y(n)=17
5: NEXT n
  40 INPUT AT 22,0;AT 0,0;"funkc
ja ? f(x,y)="; LINE f$'"zakres x
 ?"'"od : ";zx,"do: ";sx'"zakres
 y ?"'"od :";zy,"do:";sy'"zakres

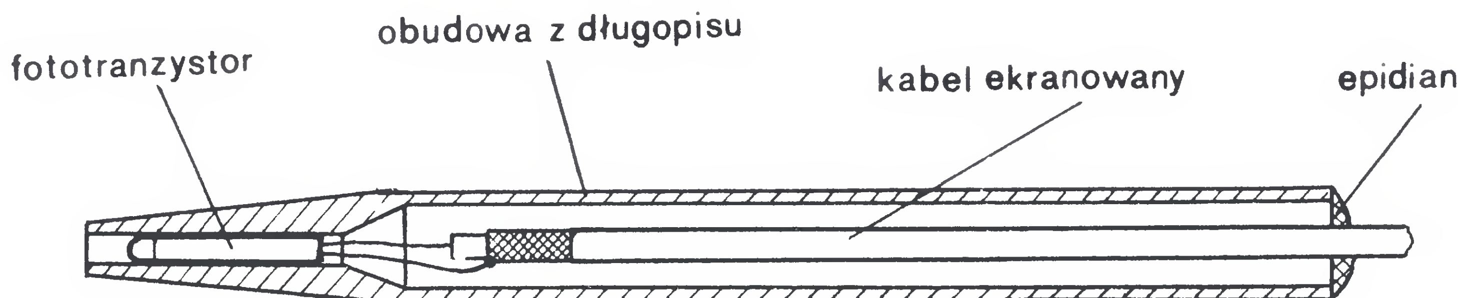
z?"'"od :";zz,"do :";sz'"dziedzi
na? (dla x i y rzeczywis-tych)";

LINE d$''" czy dane O.K.?"; LINE
 y$
  50 IF y$<>"" THEN  IF y$(1)="n
" OR y$(1)="N" THEN  GO TO 40
  60 PAPER 1: CLS
  70 PRINT AT 0,0;"z=";f$;AT 1,0
;zx;"<x<";sx;AT 2,0;zy;"<y<";sy;
AT 3,0;zz;"<z<";sz
  80 LET sx=(sx-zx)/170: LET sy=
(sy-zy)/85: LET sz=90/(sz-zz)
  90 IF d$="" THEN  LET d$="1"
 100 FOR n=0 TO 170 STEP 5
 110 PLOT n,0: PLOT n+85,85
 120 NEXT n
 130 FOR n=0 TO 90 STEP 5
 140 PLOT 0,n
 150 NEXT n
 160 FOR n=0 TO 85 STEP 3
 170 PLOT n+170,n: PLOT n,n
 180 NEXT n
 200 PRINT  INVERSE 1;AT 9,0;"z"
;AT 21,22;"x";AT 12,31;"y"
 210 FOR m=0 TO 85 STEP 2
 220 LET y=m*sy+zy: LET u=0
 230 FOR n=1 TO 170
 240 LET i=n+m: LET x=n*sx+zx
 250 IF NOT VAL d$ THEN  LET u=0
: GO TO 300
 260 LET z=(VAL f$-zz)*sz+m
 270 IF (z(i)>y(i) AND z<z(i) AN
D z>y(i)) OR z<m THEN  LET u=0:
GO TO 300
 275 IF z>m+90 THEN : LET u=0: G
O TO 301
 280 IF u THEN  DRAW 1,z-PEEK 23
678: GO TO 300
 290 LET u=1: PLOT i,z
 300 IF z>z(i) THEN  LET z(i)=z
 310 IF z<y(i) THEN  LET y(i)=z
 320 NEXT n: NEXT m
```

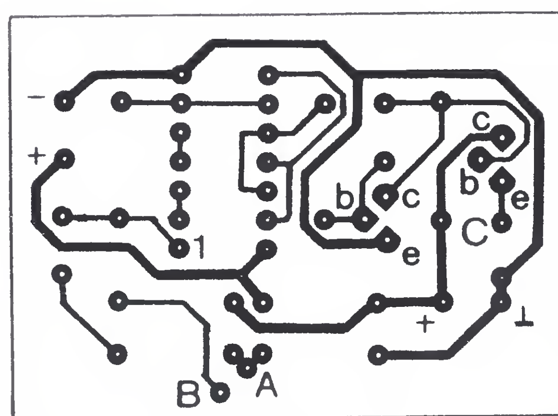
PIÓRO ŚWIETLNE DO ZX SPECTRUM



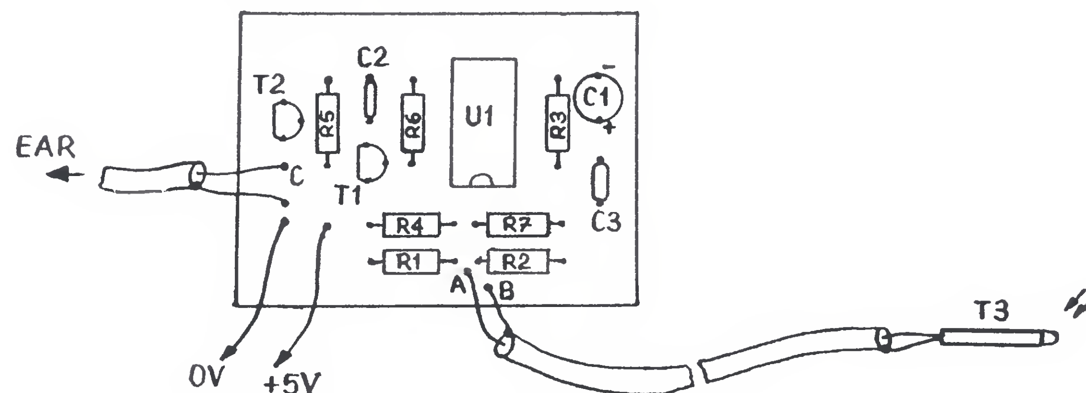
Rys. 1 — Schemat ideowy pióra świetlnego.



Rys. 2 — Konstrukcja mechaniczna pióra.



Rys. 3 — Płytka drukowana wzmacniacza sygnału.



Rys. 4 — Rozmieszczenie elementów na płytce.

Jednym z ciekawszych urządzeń, które użytkownicy komputera ZX SPECTRUM mogą tanim kosztem wykonać w domowym warsztacie elektronicznym jest pióro świetlne.

Przyrząd ten wykrywa bieg rozświetlonej plamki na ekranie (zgodnie z zasadą tworzenia obrazu telewizyjnego) i odczytuje jej położenie. Zapamiętane współrzędne plamki odpowiadają miejscu przyłożenia pióra do ekranu a to co dalej się stanie zależy już od inwencji programisty. Pióro może być pomocne przy wyborze opcji przedstawionych w „menu" na ekranie, może służyć do rysowania jakichś figur itp.

Podstawową częścią pióra świetlnego jest fototranzystor, reagujący na zmiany jasności ekranu. Prąd płynący przez fototranzystor jest następnie wzmacniany we wzmacniaczu o wysokiej oporności wejściowej wykonanym ze zlinearyzowanych inwerterów **CMOS**. Ukształtowane na wyjściu prostokątne impulsy przesyłane są na wejście **„EAR"** komputera, z którego odczyt dokonywany jest na tej samej zasadzie jak przy odczycie danych z magnetofonu. Cały układ jest zasilany napięciem +5 V pobieranym z szyny wyjściowej komputera (styk numer 3B, od strony druku). Wartość napięcia zasilającego nie jest krytyczna i można tu zastosować baterię 9 V, unikając zakupu złącza krawędziowego. Schemat urządzenia pokazany jest na rys. 1. Na rys. 2 pokazano konstrukcję pióra, zaś na rysunkach 3 i 4 płytkę drukowaną i sposób montażu elementów. Połączenie pióra ze wzmacniaczem należy wykonać za pomocą przewodu ekranowego, a wzmacniacz zamknąć w dowolnej obudowie z tworzywa sztucznego. Wykonanie urządzenia i jego uruchomienie nie powinno stwarzać kłopotów. Czytelnikom mniej doświadczonym przypominamy o zasadach obchodzenia się z układami scalonymi wykonanymi w technologii CMOS: **należy je przechowywać ze zwartymi nóżkami (np. za pomocą folii aluminiowej) i lutować za pomocą lutownicy z uziemionym grotem.**

Autorzy wykonali kilka egzemplarzy omawianego tu pióra, które działają bez zarzutu. Kłopoty przy uruchomieniu mogą być związane z rozrzutami parametrów polskich układów scalonych typu **MCY 74069** ze zbyt niskim wzmocnieniem fototranzystora. Przy uruchamianiu należy też dobrać jasność ekranu. Jeśli układ działa prawidłowo, to po dotknięciu piórem jasnego ekranu w głośniku komputera powinien być słyszalny dźwięk podobny do przydźwięku sieci, związany z okresowym wytwarzaniem obrazu telewizyjnego.

Równie ważnym jak samo pióro jest program jego obsługi. Sprawdza się tu zasada, że sprzęt bez odpowiedniego oprogramowania jest wart niewiele. Czytelnicy mogą skorzystać z gotowych programów jakie można znaleźć na polskim rynku. Przy poszukiwaniu programu warto skorzystać z oferty klubów mikrokomputerowych lub z pomocy znajomych.

Doświadczeni programiści, którzy znają system operacyjny mikrokomputera SPECTRUM, mogą pokusić się sami o napisanie programu obsługi pióra świetlnego. Program powinien odczytywać stan wejścia **„EAR"**, obliczyć opóźnienie od momentu pojawienia się pierwszego impulsu związanego z umownym „początkiem" kadru telewizyjnego do momentu przyjścia następnego impulsu związanego z konkretnym punktem ekranu, przeliczać to opóźnienie na współrzędne ekranu i zapamiętywać je. Prostsze jest napisanie programu znajdującego numer linii (jak dla instrukcji **PRINT**), w której znajduje się pióro. Przykładowo podajemy, że w firmowym programie **dK'TRONICS** podprogram ten ma długość tylko 96 bajtów, zaś podprogram obliczający współrzędne położenia pióra jak dla komendy **PLOT** aż 1706 bajtów i jest to miarą trudności przy pisaniu programu.

Na zakończenie jeszcze ciekawostka: jeden z autorów mając kłopoty z wczytywaniem danych z magnetofonu zastosował z powodzeniem opisywany tu układ wzmacniacza do standaryzacji sygnału wejściowego. Aby wzmacniacz pracował poprawnie należy zastosować rezystor R1 o wartości około 20 kΩ i doprowadzić sygnał z przedwzmacniacza w magnetofonie między punkt A i masę. Pióro świetlne należy oczywiście odłączyć od punktów A i B. Po zastosowaniu odpowiedniego przełącznika otrzymujemy uniwersalny układ, który może służyć dwom różnym celom. Koszt elementów nie przekracza 600 zł (co na to różne firmy sprzedające pióra świetlne za kilkadziesiąt tysięcy złotych?).

Wykaz części:

U1 — układ scalony **MCY 74069** (**CD4069**),
T1, **T2** — tranzystory **BC 238C** lub podobne,
T3 — fototranzystor **BPYP 21**,
C1 — kondensator elektrolityczny 100μ F/10 V,
C2 — kondensator ceramiczny 120 pF,
C3 — kondensator 100 nF (np. styroflekssowy),
rezystory: R1 — 2,4 MΩ, R2 — 10 kΩ, R3 — 510 kΩ... 1 MΩ, R4 — 100Ω, R5, R7 — 1 kΩ, R6 — 39 kΩ,
wtyk słuchawkowy „jack" 3,5 mm,
złącze krawędziowe 2×8 styków (opcjonalnie).

Konrad Fedyna
Zygmunt Wereszczyński

PROGRAM OBSŁUGI PIÓRA ŚWIETLNEGO

Przykładową realizację programową obsługi pióra, umożliwiającą wybór zaprogramowanych opcji, przedstawiono na wydruku. Pozwala on wykorzystać pióro świetlne wzorując się na popularnej „myszy". Podstawową część programu stanowią linie od 9000 do 9120. W części tej ładowany jest kod maszynowy obsługujący port 254, do którego podłączone jest pióro i obliczający, w którym z 21 wierszy na ekranie jest umieszczone. Numer zdekodowanego wiersza znajduje się w zmiennej „a" (linia 100). Wykorzystanie informacji tam zawartej jest już proste lecz wymaga pomysłowości i inwencji od piszącego program. Linie programu od 10 do 140 prezentują sposób realizacji trzech przykładowych procedur wybieranych piórem świetlnym. Linia 9999 pozwala zapisać program na taśmie magnetofonowej umożliwiając jego autostart po ponownym wczytaniu do mikrokomputera.

(kf, zw)

```
   1 REM ****************
   2 REM * KF & ZW Lpen *
   3 REM *    © 1986    *
   4 REM ****************
  10 BORDER 0: PAPER 0: INK 7:
     CLS : LET s=0
  20 PRINT AT 2,20;"kasowanie  ";
     INVERSE 1;" "
  30 PRINT AT 4,20;"prostokat  ";
     INVERSE 1;" "
  40 PRINT AT 6,20;"okrag      ";
     INVERSE 1;" "
 100 LET a=USR 64103
 110 IF a=1 THEN  RUN : GO TO 20
 120 IF a=3 THEN  LET s=s+1:
      PLOT 50+s,50: DRAW 90,0:
      DRAW 0,-30: DRAW -90,0:
      DRAW 0,30: GO TO 100
 130 IF a=5 THEN  LET s=s+1:
      CIRCLE 50+s,50,40
 140 GO TO 100
9000 CLEAR 64067
9010 FOR i=64068 TO 64120
9020 READ k: POKE i,k
9030 NEXT i: RUN
9040 DATA 14,254,33,0,0,17,88
9050 DATA 2,118,27,122,179,32
9060 DATA 251,62,255,17,0,1,237
9070 DATA 64,203,112,32,9,6,13
9080 DATA 16,254,35,27,186,32
9090 DATA 241,201,205,68,250
9100 DATA 62,0,188,32,248,203
9110 DATA 61,203,61,203,61
9120 DATA 229,193,201,0
9999 SAVE "Lpen" LINE 9000
```

BETA BASIC

NOWE MOŻLIWOŚCI ZX SPECTRUM

BETA BASIC v. 1.8 dodaje 30 nowych komend i 21 nowych funkcji do już istniejących w ZX SPECTRUM. Oprócz tego niektóre z istniejących komend zostały ulepszone, dodano też szereg zmian mających na celu ułatwienie pracy z komputerem (np. migający kursor bieżącej linii, pełne działanie klawisza BREAK czy też możliwość przesuwania kursora we wszystkich kierunkach podczas edycji linii programu).

Nowe komendy uzyskuje się po przejściu w tryb **GRAPHIC** i naciśnięciu odpowiedniego klawisza. Pojawia się wówczas odpowiednie nowe kluczowe słowo z zestawu komend **BETA BASIC**.

Nowe funkcje otrzymuje się po wprowadzeniu normalnego słowa kluczowego **FN** a następnie odpowiedniej litery i znaku **„$"** lub **„("**. Pojawia się wówczas nazwa nowej funkcji i należy podać jej argumenty.

Ładowanie systemu **BETA BASIC** odbywa się normalnie, tzn. po komendzie **LOAD ""** lub **LOAD "Beta Basic"**. Program składa się z dwóch części, z których pierwsza zawiera trzy linie o numerach 0, 1 i 2. Druga część to właściwy program w kodzie maszynowym. Po załadowaniu całości linie 1 i 2 skasują się automatycznie, zaś linia 0, zawierająca definicje funkcji **BETA BASIC** pozostanie niekasowalna. Od tej chwili system jest gotów do pracy.

OPIS KOMEND BETA BASIC

W poniższym opisie podano słowa kluczowe w kolejności alfabetycznej wraz z opisem argumentów. Argumenty występujące w nawiasach trójkątnych podaje się opcjonalnie. Podano też w nawiasie kwadratowym, pod którym klawiszem „ukryte" jest dane słowo kluczowe.

ALTER ‹opis-atrybutów› TO opis-atrybutów [A] — Komenda ta pozwala na szybką zmianę atrybutów ekranu (**INK, FLASH, BRIGHT, PAPER**) bez potrzeby oczyszczania ekranu. Przykłady: **ALTER TO PAPER 1, INK 6** lub **ALTER INK 7 TO PAPER 2, INK 0**

AUTO ‹numer-linii› ‹,krok› [6] — Włączenie automatycznej numeracji linii poczynając od danego numeru i z danym krokiem (jeśli podano) lub od linii bieżącej z krokiem 10. Wyłączenie — przytrzymanie klawisza BREAK dłużej niż 1 sekundę.

BREAK [SHIFT-SPACE] — Nie jest to słowo kluczowe, nie używa się go w trybie GRAPHICS. Umożliwia przerwanie działania każdego programu, bowiem system **BETA BASIC** pracuje w trybie **INTERRUPT 2** mikroprocesora Z80.

CLOCK liczba lub łańcuch [C] — Komenda ta steruje 24-godzinnym zegarem, który spełnia różne funkcje, zależnie od podanego argumentu:

ARGUMENT	ALARMOWY SKOK DO LINII	ALARM DŹWIĘKOWY	WYŚWIETLANIE
0	NIE	WYŁĄCZONY	BRAK
1	NIE	WYŁĄCZONY	JEST
2	NIE	USTAWIONY	BRAK
3	NIE	USTAWIONY	JEST
4	TAK	WYŁĄCZONY	BRAK
5	TAK	WYŁĄCZONY	JEST
6	TAK	USTAWIONY	BRAK
7	TAK	USTAWIONY	JEST

Wprowadzenie czasu: **CLOCK "09:29:05"**
Ustawienie czasu alarmu: **CLOCK "A06:20"**
Alarmowy skok do linii: **CLOCK "A9000"**

CHR$ 8 — kursor w lewo
CHR$ 9 — kursor w prawo
CHR$ 10 — kursor w dół
CHR$ 11 — kursor w górę

Znaki te wprowadzone do łańcucha drukowanego za pomocą **PRINT** zmieniają pozycję drukowania zgodnie z opisem obok.

DEF KEY łańcuch-jednoznakowy; łańcuch SHIFT 1
DEF KEY łańcuch-jednoznakowy: instrukcja: instrukcja:...
Podstawienie pod klawisz z podanym znakiem danego łańcucha lub ciągu instrukcji.

DEF PROC nazwa-procedury [1] — Słowo kluczowe rozpoczynające definicję procedury wywoływanej przez nazwę.

DELETE ‹numer-linii› TO ‹numer-linii› [7] — Kasowanie linii programu w podanym zakresie. Dopuszcza się stosowanie różnych postaci komendy, np. **DELETE TO 50**, **DELETE 10 TO 50**, **DELETE TO** itp.

DO
DO WHILE warunek
DO UNTIL warunek [D] — Słowo kluczowe rozpoczynające definicję pętli. Pętla musi się kończyć słowem **LOOP**.

DPOKE adres, liczba [P] — Załadowanie dwóch kolejnych bajtów pamięci o podanym adresie liczbą z zakresu od 0 do 65535.

EDIT ‹numer-linii› [0] — Wprowadzane NORMALNIE umożliwia edycję dowolnej linii. Pojawia się po naciśnięciu klawisza 0 jeśli poprzednio użyto klawisza ENTER.

ELSE instrukcja [E] — Jest to część struktury **IF-THEN** jako alternatywa instrukcji podstawowej. Sposób użycia: **IF** warunek **THEN** instrukcja **ELSE** instrukcja

END PROC [3] — Słowo kończące definicję procedury wywoływanej przez nazwę.

EXIT IF warunek [I] — Warunkowe wyjście z pętli **DO-LOOP**.

FILL x,y
FILL ‹INK kolor› ;x,y
FILL ‹PAPER kolor› ;x,y [F] — Wypełnienie obszaru tła kolorem znaku (jeśli użyto **FILL** lub **FILL INK**) lub obszaru znaku kolorem tła (jeśli użyto **FILL PAPER**). Dopuszcza się też używanie konstrukcji złożonych, np. **FILL INK 2; PAPER 1; FLASH 1;x,y**. Zapełnianie rozpoczyna się od punktu o współrzędnych x,y.

GET zmienna-numeryczna lub zmienna-łańcuchowa [G] — Nadaje zmiennej numer klawisza (z klawisza liczbowego oraz 11 dla A, 12 dla B, 13 dla C itd.) albo znak na naciśniętym klawiszu.

JOIN ‹numer-linii› [SHIFT 6] — Łączy linię o podanym numerze (lub linię bieżącą jeśli numer nie został podany) z linią znajdującą się w dolnej części ekranu, nadając nowej linii podany numer (lub numer linii bieżącej).

KEYIN łańcuch [SHIFT 4] — Dopuszcza się użycie tylko jako instrukcji w programie, powoduje ono wprowadzenie do programu podanego łańcucha (może to być np. linia programu).

KEYWORDS 1
KEYWORDS 0 [8] — Przełącznik słów kluczowych B.B. na znaki graficzne otrzymywane normalnie w trybie GRAPHICS. Słowo to nie jest nigdy wyłączane.

LIST numer-linii TO numer-linii
LLIST numer-linii TO numer-linii — Rozszerzenie syntaktyki normalnego słowa kluczowego SPECTRUM BASIC, wprowadzana NORMALNIE.

LOOP
LOOP UNTIL warunek
LOOP WHILE warunek [L] — Część struktury DO-LOOP kończąca pętlę.

ON [O] — Stosowane w strukturach postaci: **GO TO ON zmienna;numer-linii, numer-linii...** lub: **GO SUB ON zmienna;numer-linii, numer-linii...** Pozwala na skok do odpowiedniej linii w zależności od wartości zmiennej.

ON ERROR numer-linii [N] — Włączenie obsługi błędów przez linię o podanym numerze (do niej nastąpi skok po pojawieniu się błędu w czasie wykonywania programu). Dodatkowo nadaje zmiennej ERROR wartość równą kodowi błędu. Nie działa przy kodach błędów 0 (OK) i 9 (STOP). Wyłączenie obsługi błędów następuje po odbytym skoku lub po komendzie ON ERROR 0.

PLOT x,y;łańcuch [Q] — Wprowadzane NORMALNIE pozwala na **pisanie** w dowolnym miejscu ekranu (współrzędne dotyczą lewego górnego rogu pierwszego znaku łańcucha). W łańcuchu można stosować znaki sterujące kursorem.

POKE adres,łańcuch — NORMALNE słowo kluczowe o rozszerzonej syntaktyce. Powoduje wpis do pamięci podanego łańcucha znaków, poczynając od danego adresu.

POP ‹zmienna-numeryczna› [Q] — Usuwa adres ze stosu **GO SUB**, **DO-LOOP**, **PROC**. Wskazywany przez ten adres numer linii jest podstawiany jako wartość zmiennej (jeśli jej nazwa została podana).

PROC nazwa [2] — Wywołanie procedury o podanej nazwie.

RENUM ‹początek TO koniec› ‹LINE nowy-początek› ‹STEP krok› [4] — Przenumerowanie linii z podanego zakresu według podanych parametrów. Jeżeli nie został podany żaden parametr to jako początkowy numer linii przyjmuje się 10 a krok równy 10.

ROLL kod-kierunku ‹,pixele› ‹;x,y;szerokość,wysokość› [R] — „Przewijanie" zdefiniowanego okna ekranu, obraz znikający z jednej strony pojawia się po stronie przeciwnej. Podana szerokość okna dotyczy ilości pozycji znaków (a nie elementów obrazu — pixeli). Współrzędne x i y pokazują lewy górny róg wybranego okna. Komenda powoduje przesuw o jeden pixel. Jeśli podany jest tylko kod kierunku to „przewijany" jest cały ekran. Można „przewijać" cały obraz łącznie z atrybutami, tylko atrybuty lub tylko treść obrazu. Przy „przewijaniu" atrybutów podana liczba pixeli musi być równa 8. Kody kierunków mają następujące znaczenie:

KOD KIERUNKU	KIERUNEK	OBEJMUJE
1	LEWO	ATRYBUTY
2	DÓŁ	ATRYBUTY
3	GÓRA	ATRYBUTY
4	PRAWO	ATRYBUTY
5	LEWO	TREŚĆ OBRAZU
6	DÓŁ	TREŚĆ OBRAZU
7	GÓRA	TREŚĆ OBRAZU
8	PRAWO	TREŚĆ OBRAZU
9	LEWO	CAŁOŚĆ
10	DÓŁ	CAŁOŚĆ
11	GÓRA	CAŁOŚĆ
12	PRAWO	CAŁOŚĆ

SCROLL ‹kod-kierunku› ‹,pixele› ‹;x,y;szerokość, wysokość› [S] — Przesuw zawartości zdefiniowanego okna ekranu o jeden pixel. Syntaktyka jak w **ROLL**, z tym że można nie podać żadnych parametrów i wówczas nastąpi przesuw całego ekranu o 1 linię do góry (jak w ZX-81). Zawartość ekranu znika na krawędzi okna.

KLAN SPECTRUM



SORT tablica lub łańcuch
SORT INVERSE tablica lub łańcuch [M] — Porządkuje tablicę znakową lub łańcuch w kolejności kodów znaków (lub odwrotnie) oraz tablicę liczbową w kolejności od największej liczby do najmniejszej (lub odwrotnie). Tablica liczbowa może mieć najwyżej 2 wymiary.

SPLIT — Nie jest to słowo kluczowe, zamiast niego wprowadza się NORMALNIE znak "‹›" (SYMBOL SHIFT W). Wskazuje on, w którym miejscu linia programu ma być rozdzielona na dwie części. Część przed tym znakiem jest dołączana do programu, zaś na dole pozostaje reszta linii z tym samym numerem.

TRACE ‹numer-linii› [T] — Komenda powodująca krokowe wykonanie programu, jej podanie powoduje skok **GO SUB** do podanej linii bezpośrednio przed wykonaniem każdej instrukcji programu. Wyłączane przez **RUN**, **CLEAR** i **TRACE 0**.

UNTIL warunek [K] — Część pętli **DO-LOOP** używana w **DO UNTIL** lub w **LOOP UNTIL**, pozwalająca na jej warunkowe wykonanie.

USING łańcuch-wzorzec;liczba [P] — Stosowane w instrukcji PRINT USING pozwala nadawać odpowiedni format wydrukowi liczb. Znaki "#" w łańcuchu-wzorcu oznaczają spacje poprzedzające.

WHILE warunek [J] — Część pętli **DO-LOOP** używana w **DO WHILE** lub **LOOP WHILE**. Pozwala na warunkowe wykonanie pętli.

OPIS FUNKCJI SYSTEMU BETA BASIC

Poniżej podano krótki opis funkcji dostarczanych przez system **BETA BASIC** wraz z opisem ich argumentów oraz sposobem wprowadzania (w nawiasie kwadratowym).

AND (liczba,liczba) [FN A() — Iloczyn logiczny dwóch liczb wykonany na ich reprezentacji binarnej.
BIN$ (liczba) [FN B$() — Binarny odpowiednik liczby dziesiętnej.
CHAR$ (liczba) [FN C$() — Konwersja liczby całkowitej bez znaku z przedziału 0 — 65535 w równoważny łańcuch dwuznakowy.
COSE (liczba) [FN C() — Cosinus liczby (4 miejsca znaczące, szybszy niż w oryginalnym systemie SPECTRUM).
DEC (łańcuch) [FN D() — Dziesiętny odpowiednik liczby w zapisie heksadecymalnym.
DPEEK (adres) [FN P() — Wartość dwubajtowej liczby znajdującej się pod podanym adresem.
FILLED () [FN F() — Ilość elementów obrazu zapełniona przez ostatnią komendę **FILL**.
HEX$ (liczba) [FN H$() — Heksadecymalny równoważnik liczby w zapisie dziesiętnym.
INSTRING (start,łańcuch1,łańcuch2) [FN I() — Pozycja pierwszego znaku łańcucha 2 w łańcuchu 1 przy przeszukiwaniu łańcucha 1 od podanej pozycji startowej lub zero, jeśli łańcuch 2 nie występuje w łańcuchu 1.
MEM () [FN M() — Ilość wolnych bajtów pamięci.
MEMORY$ () [FN M$() — Zawartość całej pamięci (od adresu 1 do 65532) traktowana jako łańcuch.
MOD (liczba1,liczba2) [FN V() — Wynik dzielenia liczby 1 modulo liczba 2.
NUMBER (łańcuch) [FN N() — Liczba całkowita bez znaku z przedziału 0 — 65535 będąca odpowiednikiem dwuznakowego łańcucha.
OR (liczba1,liczba2) [FN O() — Suma logiczna dwóch liczb wykonana na ich zapisie bitowym.
RNDM (liczba) [FN R() — Liczba pseudolosowa z przedziału od zera do podanej liczby włącznie.
SCRN$ (wiersz,kolumna) [FN K$() — Znak znajdujący się na podanej pozycji (także znak zdefiniowany przez użytkownika).
SINE (liczba) [FN S() — Sinus danej liczby (4 miejsca znaczące).
STRING$ (liczba,łańcuch) [FN S$() — Powtórzenie łańcucha podaną ilość razy.
TIME$ () [FN T() —Bieżący czas, mierzony przez **CLOCK**.
USING$ (łańcuch-wzorzec,liczba) [FN U$() — Znakowy zapis liczby w podanym formacie (jak USING).
XOR (liczba1,liczba2) [FN X() — Logiczna operacja **EXCLUSIVE OR** na bitowym zapisie dwóch liczb.

ZMIENNE SPECJALNE

W systemie **BETA BASIC** istnieją pewne zmienne generowane przez system i dostępne przez nazwę. Poniżej podano ich opis:
XOS,YOS — współrzędne środka układu odniesienia (ustawione początkowo na 0,0). Można je zmieniać w zakresie od 0 do 255 dla XOS i od 0 do 175 dla YOS. Zerowane przez komeny CLEAR i RUN.
XRG,YRG — zakres współrzędnych na ekranie (początkowe wartości: 256 dla XRG i 176 dla YRG), który można dowolnie zmieniać. **CLEAR** i **RUN** przywracają wartości początkowe.
Podczas wykonywania komendy ON ERROR lub TRACE tworzone są zmienne specjalne o nazwach:
ERROR — podaje kod błędu, który był ostatnio wykryty.
LINE — numer ostatnio wykonanej linii (przy **TRACE**) lub linii, w której wystąpił błąd (przy **ON ERROR**).
STAT — numer ostatnio wykonanej instrukcji (przy **TRACE**) lub instrukcji, w której wystąpił błąd (przy ON ERROR).
Nazwy zmiennych specjalnych można wprowadzać przy użyciu małych lub wielkich liter.
System BETA BASIC otwiera nowe, wspaniałe możliwości programowania, lecz dojście do wprawy w posługiwaniu się tym narzędziem wymaga dużo ćwiczeń. Nie należy się jednak zrażać, bo efekty, które można osiągnąć są znakomite.
W jednym z najbliższych numerów postaramy się przedstawić nową wersję systemu BETA BASIC v. 3.0. i v. 3.1.

Konrad Fedyna
Zygmunt Wereszczyński

ZEGAR • ZEGAR • ZEGAR

1. Podstawowe własności i sposób posługiwania się zegarem:

- zegar przystosowany jest do współpracy z interpreterem BASIC-u i działa „równolegle" z nim.
- ładowanie zegara z kasety odbywa się przez operację pulpitową: należy wcisnąć klawisz START i włączyć komputer. Po załadowaniu powinien zgłosić się interpreter BASIC-u.
- zegar posiada wbudowaną możliwość ustawiania daty i czasu za pomocą klawiszy konsoli.
  W tym celu należy wcisnąć na ok. 5 sekund START. Gdy zaczną migać sekundy zegar jest gotów do ustawienia. Klawiszem SELECT dokonuje się wyboru ustawionych cyfr, a samo przestawienia, klawiszem OPTION. Ponowne naciśnięcie START powoduje powrót do normalnego trybu pracy.
- zegar znika z ekranu (ale działa dalej) po wykonaniu instrukcji GRAPHICS. Można spowodować ponowne jego wyświetlanie przez instukcję.
  X = USR (1633).
- Jedyną reakcją programu na wciśnięcie RESET jest chwilowe wstrzymanie jego pracy.
- W czasie transmisji (do kasety wyświetlanie czasu zostaje wstrzymane), aby nie przeszkadzać procedurom systemu operacyjnego.
- zegar ma możliwość regulacji. Można to uczynić na dwa sposoby

a) w czasie jego działania zmienić zawartość bajtów 1603 (młodszy) i 1604 (starszy). Zmniejszenie zawartości tych komórek powoduje przyspieszenie zegara.
b) W instrukcji DATA nr. 119 liczby 104, 1 odpowiadają opisanym wyżej bajtom.

Po ich modyfikacji należy obliczyć i zmienić sumę kontrolną (ostatnia liczba w każdej instrukcji) DATA, oraz wygenerować zegar na nowo.

2. Wygenerowanie programu zegara.

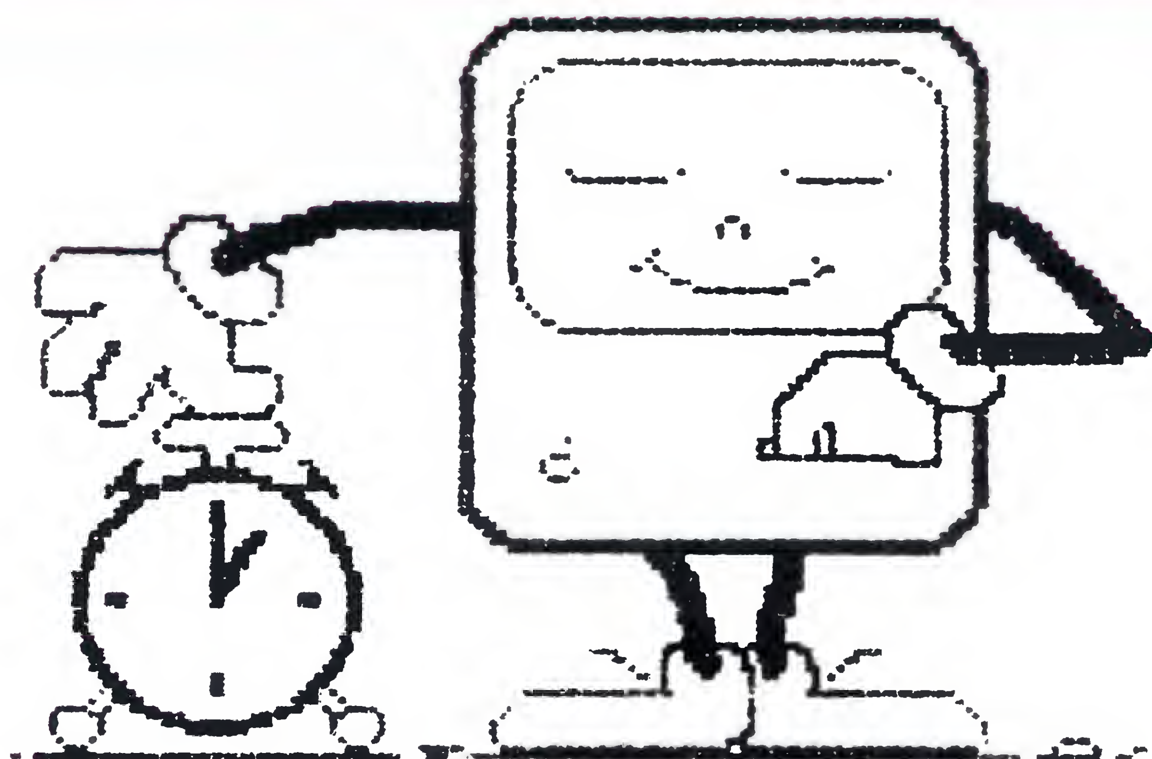
Program napisany został początkowo w języku assemblera i skompilowany. W celu udostępnienia go również tym, którzy nie posiadają kompilatora został przetłumaczony na ciąg instrukcji DATA (linie 111–200 programu). Instrukcje 101–110 zawierają procedurę służącą do zapisania programu zegara na kasecie.

Ostatnią liczbą w każdej instrukcji jest suma kontrolna, która minimalizuje prawdopodobieństwo popełnienia błędu.

Na koniec lista czynności, którą należy wykonać, aby otrzymać gotowy program zegara.

- przepisać dokładnie umieszczony obok tekst programu.
- nagrać go (może się jeszcze przydać).
- uruchomić program, przygotować kasetę, na której ma być nagrany zegar.
- jeżeli wszystko przebiega poprawnie, to po usłyszeniu podwójnego sygnału dźwiękowego należy włączyć zapis, a po chwili program będzie gotów.
- jeśli zdarzył się błąd — nie należy się poddawać, sprawdzić poprawność, wprowadzonego tekstu, i spróbować jeszcze raz. Na pewno się uda.

Marek Markowski



```
10 REM LADOWACZ ZEGARA
15 N=100:GOSUB 55
20 FOR I=32768 TO 33567
25 READ A:POKE I,A:CS=CS+A:L=L+1
30 IF L=9 THEN GOSUB 45
35 NEXT I
40 X=USR(32768):END
45 READ CSD
50 IF CSD<>CS THEN GOSUB 60
55 CS=0:L=1:N=N+1:RETURN
60 ? "BLAD W WIERSZU ";N;" lub ";N-1
65 LIST N-1,N:STOP
99 REM ZEGAR
100 REM Marek Markowski 1986.08.04
101 DATA 162,16,169,3,157,66,3,169,745
102 DATA 8,157,74,3,169,128,157,75,771
103 DATA 3,169,72,157,68,3,169,128,769
104 DATA 157,69,3,32,86,228,48,36,659
105 DATA 169,11,157,66,3,169,80,157,812
106 DATA 68,3,169,128,157,69,3,169,766
107 DATA 0,157,72,3,169,3,157,73,634
108 DATA 3,32,86,228,48,8,169,12,586
109 DATA 157,66,3,32,86,228,104,96,772
110 DATA 67,58,158,0,0,0,0,0,283
111 DATA 0,6,0,6,98,6,169,60,345
112 DATA 141,2,211,169,218,141,231,2,1115
113 DATA 133,14,169,8,141,232,2,133,832
114 DATA 15,173,254,191,133,10,173,255,1204
115 DATA 191,133,11,24,96,50,41,50,596
116 DATA 49,50,49,50,50,49,17,14,328
117 DATA 11,1,4,7,50,49,50,247,419
118 DATA 7,30,8,47,8,109,8,82,299
119 DATA 8,64,8,104,1,112,112,70,479
120 DATA 77,6,1,0,0,0,24,22,130
121 DATA 14,16,24,14,16,20,0,0,104
122 DATA 16,16,26,16,16,26,16,16,148
123 DATA 0,104,169,144,141,34,2,169,763
124 DATA 6,141,35,2,24,173,48,2,431
125 DATA 133,204,105,3,141,75,6,173,840
126 DATA 49,2,133,205,141,76,6,160,772
127 DATA 0,169,1,145,204,200,169,69,957
128 DATA 145,204,200,169,6,145,204,96,1169
129 DATA 248,162,0,238,197,8,208,3,1064
130 DATA 238,198,8,238,199,8,173,199,1261
131 DATA 8,201,50,208,12,142,199,8,828
132 DATA 24,173,200,8,105,1,141,200,852
133 DATA 8,173,197,8,205,67,6,208,872
134 DATA 16,173,198,8,205,68,6,208,882
135 DATA 8,142,197,8,142,198,8,240,943
136 DATA 210,165,66,208,13,173,200,8,1043
137 DATA 240,22,32,149,7,32,202,7,691
138 DATA 208,10,173,200,8,201,16,48,864
139 DATA 3,32,149,7,216,76,226,192,901
```

```
140 DATA 173,208,8,208,25,169,1,44,836
141 DATA 31,208,208,13,238,209,8,208,1123
142 DATA 235,141,208,8,141,209,8,240,1190
143 DATA 85,142,209,8,208,222,32,166,1072
144 DATA 8,24,169,1,44,31,208,208,693
145 DATA 19,142,210,8,142,211,8,173,913
146 DATA 209,8,208,117,32,184,8,142,908
147 DATA 208,8,208,109,42,44,31,208,858
148 DATA 208,55,142,209,8,142,211,8,983
149 DATA 173,210,8,240,14,201,25,240,1111
150 DATA 5,238,210,8,208,83,169,15,936
151 DATA 141,210,8,32,184,8,173,208,964
152 DATA 8,201,6,208,3,142,208,8,784
153 DATA 238,208,8,238,210,8,172,208,1290
154 DATA 8,185,45,6,141,212,8,208,813
155 DATA 48,42,44,31,208,208,33,142,756
156 DATA 209,8,142,210,8,173,211,8,969
157 DATA 240,14,201,25,240,5,238,211,1174
158 DATA 8,208,22,169,15,141,211,8,782
159 DATA 32,143,8,238,211,8,208,9,857
160 DATA 142,209,8,142,210,8,142,211,1072
161 DATA 8,216,76,226,192,142,200,8,1068
162 DATA 32,10,8,48,23,32,30,8,191
163 DATA 48,18,32,47,8,48,13,32,246
164 DATA 64,8,48,8,32,82,8,208,458
165 DATA 3,32,109,8,96,72,74,74,468
166 DATA 74,74,9,16,153,77,6,200,609
167 DATA 104,41,15,9,16,153,77,6,421
168 DATA 200,96,160,1,173,204,8,32,874
169 DATA 181,7,200,173,205,8,32,181,987
170 DATA 7,200,173,206,8,32,181,7,814
171 DATA 160,11,173,203,8,32,181,7,775
172 DATA 200,173,202,8,32,181,7,200,1003
173 DATA 173,201,8,32,181,7,96,142,840
174 DATA 199,8,173,201,8,201,48,48,886
175 DATA 5,169,1,32,30,8,142,201,588
176 DATA 8,96,24,109,201,8,141,201,788
177 DATA 8,201,96,48,8,56,233,96,746
178 DATA 141,201,8,169,1,96,24,109,749
179 DATA 202,8,141,202,8,201,96,48,906
180 DATA 5,142,202,8,169,1,96,24,647
181 DATA 109,203,8,141,203,8,201,36,909
182 DATA 48,5,142,203,8,169,1,96,672
183 DATA 24,109,206,8,141,206,8,205,907
184 DATA 207,8,48,5,169,1,141,206,785
185 DATA 8,96,24,109,205,8,141,205,796
186 DATA 8,201,19,48,5,169,1,141,592
187 DATA 205,8,168,185,36,6,141,207,956
188 DATA 8,152,201,1,96,24,109,204,795
189 DATA 8,141,204,8,240,12,41,19,673
190 DATA 240,4,201,18,208,4,169,48,892
191 DATA 208,2,169,41,141,38,6,173,778
192 DATA 205,8,201,2,240,212,96,173,1137
193 DATA 208,8,24,42,168,185,53,6,694
194 DATA 141,213,8,185,54,6,141,214,962
195 DATA 8,169,1,108,213,8,169,2,678
196 DATA 44,199,8,240,11,172,212,8,894
197 DATA 138,153,77,6,153,78,6,96,707
198 DATA 172,208,8,185,200,8,172,212,1165
199 DATA 8,32,181,7,96,0,0,0,324
200 DATA 0,0,0,0,134,8,4,50,196
```

Klan Atari redaguje Wiesław Migut

TEST OCZU I USZU

Ten program sprawdzi czas Twojej reakcji na nieoczekiwanie pojawiający się znak na ekranie i sygnał dźwiękowy. Podobny jest do testów psychomotorycznych, którym poddawani są kierowcy, piloci i kosmonauci.

Składa się z dwóch programów. Najpierw trzeba wpisać PROGRAM 1, który jest „okładką" i nagrać na kasetę lub dysk pod dowolnym tytułem, choćby „**TEST-O-U**". Drugi program jest wywoływany w linii 140 rozkazem RUN"!TEST" i przy nagrywaniu trzeba użyć nazwy **TEST**. Ma to znaczenie w przypadku pracy ze stacją dysków.

Fierwszy program jest kasowany, gdy do pamięci wczytywany jest program drugi, ale rysunek na ekranie pozostaje do końca wczytywania.

Instrukcje ukazują się na ekranie, a posługiwanie się TESTEM jest banalnie proste. Dotyczy to też kasowania sygnału. W ten sposób będziesz miał chwilę wytchnienia w czasie TESTU. Linie 40—70 umożliwiają uzyskanie na ekranie polskich liter. Spróbuj po ich wpisaniu jak działają klucze klawiatury numerycznej (0—9) oraz kropka, male ENTER i znak   . Będzie to potrzebne przy pisaniu instrukcji w liniach 100—200.

Natomiast literę Ł w wyrazie ŁADOWANIE, w programie 1, uzyskasz, gdy po wypisaniu L naciśniesz jednocześnie CTRL i H (daje to znak ←), a zaraz potem znak dzielenia). Ta strzałka to informacja dla komputera, żeby w jednym położeniu kursora wyświetlił dwa znaki.

Zupełnie przyzwoity wynik to czas reakcji od 0,20:0,30 sek. Komputer sam to przeliczy.

(tp)

■ *Listing pierwszy*

```
10 ' *** PROGRAM  1  ***
20 '
30 '
40 BORDER 1:INK 0,1:INK 1,24:INK 2,26:INK 3,6:MODE 1
50 ORIGIN 0,0,0,640,0,98:CLG 3
60 FOR f=-300 TO 940 STEP 50:MOVE 320,200:DRAW f,0,2:NEXT
70 f=104:s=2
80 MOVE 0,f:DRAWR 640,0:f=f-s:s=s+2:IF f>0 THEN 80
90 ORIGIN 0,0,0,640,0,400
100 RAD:FOR f=-PI/2 TO PI/2 STEP 0.02:MOVE SIN(f)*100+320,CO
S(f)*100+100:DRAWR 0,-COS(f)*100,1:NEXT
110 PRINT CHR$(22)CHR$(1):PEN 2:LOCATE  7,9:PRINT CHR$(164)"
 ADAPTACJA TOMASZ PYC 1985.":LOCATE 28,8:PRINT","
120 LOCATE 9,3:PEN 2:PRINT"  TEST  OCZU  I  USZU":PEN 3:LOCA
TE 9,5:PRINT"  ŁADOWANIE  ok.1 min."
130 MOVE 100,380:DRAWR 440,0,1:DRAWR 0,-70:DRAWR -440,0:DRAW
R 0,70
140 PRINT CHR$(22)CHR$(0):RUN"!TEST"
```

■ *Listing drugi*

```
10 REM  *****   PROGRAM 2     *****
20 REM ***** TEST OCZU I USZU *****
30 REM ---------------------------
40 SYMBOL AFTER 91:SYMBOL 192,&0,&0,&78,&C,&7C,&CC,&7E,&6:SY
MBOL 193,&C,&18,&3C,&66,&60,&66,&3C,&0:SYMBOL 194,&18,&3C,&6
6,&C0,&C0,&66,&3C,&0:SYMBOL 195,&0,&0,&3C,&66,&7E,&60,&3C,&C
:SYMBOL 196,&F0,&60,&6C,&78,&72,&66,&FE,&0

50 SYMBOL 197,&38,&1E,&1C,&38,&78,&18,&3C,&0:SYMBOL 198,&C,&
18,&98,&66,&66,&66,&66,&0:SYMBOL 199,&C,&18,&3C,&66,&66,&66,
&3C,&0:SYMBOL 200,&C,&18,&3C,&60,&3C,&6,&7C,&0:SYMBOL 201,&F
E,&86,&C,&7E,&30,&66,&FE,&0

60 SYMBOL 202,&18,&3C,&66,&70,&1C,&46,&7C,&0:SYMBOL 203,&C,&
18,&7E,&4C,&18,&32,&7E,&0:SYMBOL 204,&18,&0,&7E,&4C,&18,&32,
&7E,&0:SYMBOL 92,&0,&0,&3C,&66,&7E,&60,&3C,&C

70 KEY 135,CHR$(192):KEY 136,CHR$(193):KEY 137,CHR$(194):KEY
 132,CHR$(195):KEY 132,CHR$(196):KEY 133,CHR$(197):KEY 134,C
HR$(198):KEY 129,CHR$(199):KEY 130,CHR$(200):KEY 131,CHR$(20
1):KEY 128,CHR$(202):KEY 138,CHR$(203):KEY 139,CHR$(204)

80 RANDOMIZE TIME
90 DIM t(10)
100 PAFER 0:PEN 1: MODE 1
110 PRINT"TEN PROGRAM SPRAWDZI CZAS TWOJEJ REAKCJI":PRINT"**
*****************************************"
120 PRINT:PRINT"     PIERWSZY TEST JEST DLA OCZU,  "
130 PRINT:PRINT"                 DRUGI DLA USZU. "
140 PRINT:  PRINT" W pierwszym teście pojawia się żółta "

150 PRINT:  PRINT"    plamka gdziekolwiek na ekranie."
160 PRINT:  PRINT"      W drugim usłyszysz sygnał."
170 PRINT:  PRINT"Szybko musisz nacisnąć klucz lub spust."
180 PRINT:PRINT"Zawsze będzie 10 prób,po każdej naciskaj"
190 PRINT"KLUCZ lub SPUST.Wynik będzie zliczany..."
200 PRINT:PRINT"---------------------------------------":PR
INT"       NACISKAJ KLUCZ LUB SPUST"
210 i$=INKEY$:IF i$="" THEN 210
220 CLS:BORDER 16
230 WINDOW #0,1,40,20,25:WINDOW #1,1,40,1,19
240 REM       *** PLAMKA ***
250 a$=CHR$(143)+CHR$(143)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(143
)+CHR$(143)
260 PAPER #0,3:PEN #0,2
270 FOR u=0 TO 9
280 CLS
290 REM ** RANDOMIZE-CZAS OD 1 DO 11 SEK
300 o=INT(RND(1)*10)+1:g1=TIME
310 g2=TIME:j$=INKEY$:IF J$<>"" THEN PRINT"Za wcześnie !!!":
GOTO 300
320 IF ((g2-g1)/300)<g THEN 310
330 x=INT(RND(1)*35)+1:y=INT(RND(1)*15)+1
340 LOCATE #1,x,y:PRINT#1,A$;:t=TIME
350 i$=INKEY$:IF i$="" THEN 350
360 t(u)=(TIME-t)/300
370 PRINT"CZAS REAKCJI:";:PRINT USING"##.##";t(u)
380 i$=INKEY$:IF i$="" THEN 380
390 CLS#1
400 NEXT u
410 CLS
420 tijd=0
430 FOR x=0 TO 9:tijd=t(X)+tijd:NEXT x
440 tijd=tijd/10
450 PRINT"CZAS REAKCJI TWOICH OCZU:";:PRINT USING "##.##";ti
jd
460 REM  *** TERAZ BLOK NA TEST USZU ***

470 PRINT;TERAZ TWOJE USZY"
480 i$=INKEY$:IF i$="" THEN 480
490 FOR u=0 TO 9
500 CLS
510 g=INT(RND(1)*10)+1:g1=TIME
520 g2=TIME:j$=INKEY$:IF J$<>"" THEN PRINT"za wcześnie!!!":G
OTO 510
530 IF ((g2-g1)/300)<g THEN 520
540 PRINT#1,CHR$(7):t=TIME
550 i$=INKEY$:IF i$="" THEN 550
560 t(u)=(TIME-t)/300
570 PRINT"CZAS REAKCJI:";:PRINT USING"##.##";t(u)
580 i$=INKEY$:IF i$="" THEN 580
590 NEXT u
600 CLS
610 tijd=0
620 FOR x=0 TO 9:tijdo=t(X)+tijdo:NEXT x
630 tijdo=tijdo/10
640 PRINT"CZAS REAKCJI TWOICH USZU:";:PRINT USING "##.##";ti
jdo
650 REM ** TEN-BLOK NA TEST MIESZANY **
660 PRINT"A TERAZ KOMBINACJA OCZU I USZU"
670 i$=INKEY$:IF i$="" THEN 670
680 CLS#1
690 FOR u=0 TO 9
700 CLS#1
710 CLS
720 g=INT(RND(1)*10)+1:g1=TIME
730 g2=TIME:j$=INKEY$:IF J$<>"" THEN PRINT"Za wcześnie !!!":
GOTO 720
740 IF ((g2-g1)/300)<g THEN 730
750 x=INT(RND(1)*35)+1:Y=INT(RND(1)*15)+1
760 IF RND(1)<0.5 THEN PRINT #1,CHR$(7) ELSE LOCATE #1,x,y:P
RINT #1,a$
770 t=TIME
780 i$=INKEY$:IF i$="" THEN 780
790 t(u)=(TIME-t)/300
800 PRINT"CZAS REAKCJI:";:PRINT USING"##.##";t(u)
810 i$=INKEY$:IF i$="" THEN 810
820 NEXT u
830 CLS
840 FOR x=0 TO 9:tijdg=t(X)+tijdg:NEXT x
850 tijdg=tijdg/10
860 PRINT"CZAS REAKCJI OCZU/USZU:";:PRINT USING "#.##";tijdg
870 PRINT"CZAS REAKCJI PODCZAS TESTU:";:PRINT USING"#.##";(t
ijd+tijdo+tijdg)/3
880 PRINT"JESZCZE RAZ ?...TO NACISKAJ KLUCZ"
890 i$=INKEY$:IF i$="" THEN 890
900 IF i$=" " THEN RUN
```

MUZYKA ROBOTA

Jeśli umieścicie w swoim programie linie od 40 do 190 jako podprogram, możecie zilustrować muzycznie np. czołówkę (okładkę) zaprojektowanej gry. Należy wykorzystać przy tym instrukcje **GOSUB** i **RETURN**, a także odpowiednio do potrzeb przenumerować linie powyższego programu. Melodia łatwo „wpada w ucho".

(tp)

```
10 '* MUZYKA ROBOTA z GRY 'ROBOT RON'*
20 'TP
30 '
40 WHILE INKEY(47)<0 AND INKEY(76)<0
50 RESTORE 140:READ pitch:SOUND 129,213,
200,5:SOUND 130,253,2000,5:SOUND 1,127,2
000,5
60 WHILE pitch>0 AND INKEY(47)<0 AND INK
EY(76)<0
70 SOUND 4,pitch,10,7:SOUND 4,159,10,7:R
EAD pitch
80 WEND
90 READ pitch:SOUND 129,319,200,6:SOUND
1,213,200,5:SOUND 1,159,200,5
100 WHILE pitch>0 AND INKEY(47)<0 AND IN
KEY(76)<0
110 SOUND 4,pitch,12,7:READ pitch
120 WEND
130 WEND
140 DATA 142,127,119,142,127,119,127,119
,106,127,119,106,119,106,95,119,106,95,1
06,95,84,106,95,84,0
150 DATA 80,84,95,84,95,106,95,106,119,1
06,119,127
160 DATA 80,84,95,84,95,106,95,106,119,1
06,119,127
170 DATA 119,127,142,159,169,190,213,239
,253,284,319,253,284,319,239,253,284,213
,239,253,190,213,239,169,190,213,190.179
,169,159,142,159,0
180 '
190 WHILE INKEY$<>"":WEND:CALL &BCA7
200 'ewentualnie RETURN
```

TRANSMAT

Na polskim rynku już dość dawno (w 1985 roku) pojawił się program, umożliwiający przepisywanie zawartości taśm na dyski. Zawiera też kilka podprogramów, które służą wygodzie użytkownika.

Program uruchamia się rozkazem **RUN"TRANSMAT"**. Po wczytaniu do pamięci na ekranie ukaże się komunikat: **"TRANSMAT 1.0-(C) Pride Utilities 1985"**, a pod spodem czarny znak kursora. Oznacza to, że będziesz mógł od tej chwili wprowadzać rozkazy. **TRANSMAT** rozumie te, które podano poniżej i to wpisane **DUŻYMI LITERAMI:**

ROZKAZY

TRANS n AUTO można przepisać n programów z taśmy na dysk. Symbol n przybiera wartości z zakresu liczb 0 — 255. Jeżeli nie znasz ilości programów na taśmie, wpisuj zero. Po zakończeniu kopiowania wciśnij ESC, by wrócić do opcji wprowadzania rozkazów. Rozkazu tego używaj tylko wtedy, gdy jesteś pewien, że programy ruszą po skopiowaniu.

TRANS n Rozkaz podobny do powyższego. Po jego wpisaniu komputer będzie czekał aż wpiszesz własną nazwę. Intencją autorów programu było zapobieganie tworzeniu dwóch programów o tej samej nazwie. Wciśnięcie CLR spowoduje przepisanie programu.

INFO Wyrzuci na ekran informacje o zbiorach na dysku (odpowiednik CAT).

DISCA Wymusza współpracę ze stacją dysków A.

DISCB Wymusza współpracę ze stacją dysków B.

BASIC Umożliwia przejście do języka BASIC. (Jak powrócić do programu TRANSMAT wyjaśniono dalej).

ERA nazwa Usuwa plik lub program. Zamiast n trzeba podstawić nazwę.

REN nowa=stara Zmienia nazwę pliku lub programu. Usuwa starą nazwę i wstawia nową. Trzeba te nazwy podać po REN.

DIR Wypisuje na ekranie tytuły plików zapisanych na dysku.

USR Zmienia liczbę n.

Przy podłączonej stacji dysków HIMEM obniżone jest do 1284 B i może się zdarzyć, że ładowany do pamięci program jest zbyt długi. Powoduje to zawieszenie ładowania. **TRANSMAT** będzie usiłował wyjść z tego w sposób nieautomatyczny... Będzie podawał umiejscowienie **(LOC)**, długość **(LEN)** i adres wykonawczy **(EXC)**. Następnie zapyta czy chcesz dodać relokator. Wciśnięcie **N** będzie przenosiło program w oryginale. Wciśnięcie **Y** przesunie program do najwyższego punktu w pamięci i doda 14-bitowy relacotor do początku. Adres **(EXC)** będzie zwykle taki sam jak **(LOC)**. Odpowiada to miejscu, gdzie wykonywany jest relokator, gdy ładowany jest program. Jeśli przepisywany program posiada adres **(EXC)** wtedy wywołanie i uruchomienie relokatora przeniesie go w odpowiednie miejsce. Jeśli adresu wykonawczego brakuje, to wywołanie **(DO NOT USE RUN)** przeniesie program i wróci do BASIC-a.

Poniżej krótki program, który umożliwia powrót do programu **TRANSMAT** po wybraniu rozkazu BASIC (patrz wykaz rozkazów), a także załadowanie go, gdy masz już w pamięci jakiś program w BASIC-u:

```
10 MEMORY & 8DFF
20 LOAD "TRANSMAT.BIN"
30 CALL & 8EØØ
RUN
```

Z pomocą programu TRANSMAT możesz wydrukować listę zbiorów zawartych na dysku. W tym celu załaduj **TRANSMAT.** Gdy ukaże się znak **A›** i znak kursora wpisz **INFO.** Na ekranie pojawi się komunikat:

Insert disc then press ENTER or P for printer

Włóż dysk, a potem wciśnij ENTER lub P dla drukarki).

Po naciśnięciu **P** komputer wydrukuje listę z jednej strony dysku. Po zmianie strony dysku powtórz tę operację. Można także uzyskać wydruk zwężoną czcionką, tzw. **CONDENSE** co szczególnie jest przydatne, gdy chcemy umieścić go w pudełku. W tym celu **PRZED** wczytaniem programu TRANSMAT trzeba przeprowadzić inicjalizację drukarki. Wystarczy w tym celu napisać:

PRINT # 8,CHR$(15)

Kod o wartości 15, nakazujący drukowanie wąską czcionką jest rozumiany przez większość drukarek.

(tp)

Klan Amstrad-Schneider redagują: Tomasz Pyć, Sergiusz Wolicki

movie

Movie przenosi cię w świat mafii i społecznego marginesu NOWEGO Jorku. Stajesz się prywatnym detektywem, masz dokumenty na nazwisko Jack Peter Marlowe i zostałeś wynajęty przez gen. Ruthwena, najbogatszego człowieka w USA, do odnalezienia zaginionej KASETY. Wiesz, że KASETA znajduje się w rękach najniebezpieczniejszego człowieka w Nowym Jorku „Cappo di Tutti Cappi" — don Emmanuela Prizzi. Jego melina mieści się gdzieś w BROOKLINIE, musisz go znaleźć. Na początek do obrony masz tylko swoje pięści, ale... znajdziesz broń, pieniądze, granaty, i inne akcesoria niezbędne w tej eskapadzie. Gdzie co znaleźć powie ci mapa tj. dokładny plan z oznakowaniem miejsc „pobytu" płatnych informatorów, morderców, kobiet, potrzebnych przedmiotów. Na planie zaznaczono też miejsca, w których powinieneś być szczególnie ostrożny tzn. pułapki, niebezpieczne przejścia czy pomieszczenia z psami.

Na dole ekranu masz 10 okienek, które pozwalają kontrolować grę. Poszczególne okna wybierasz klawiszem FIRE. Okna (od lewej):

SELECT	— nastawienie obiektu do wzięcia, położenia lub rzutu;
TAKE	— wzięcie, podniesienie obiektu;
DROP	— położenie obiektu;
FIRE	— strzał (jeśli masz już pistolet);
MOVE	— ruch w 4 kierunkach (w zależności od uprzedniego wyboru DIRECTIONAL — ruch w nakazanym kierunku, lub „NORMAL" — przód, tył i obroty);
SPEECH	— dymek do dialogów (rozmowy oczywiście tylko po angielsku);
PUNCH	— cios pięścią na wysokości klatki piersiowej;
THROW	— rzut obiektem (ustawionym przez SELECT);
HALT	— przerwa w grze, pauza;
ABORT	— powrót do początku (nowa gra);

Warto wiedzieć, że podczas „obsługi" okienek klawisz w dół lub ruch drążka sterowego w dół powoduje automatyczne przejście do opcji MOVE — można od razu rozpocząć marsz.

A teraz w drogę — z pokoju do pokoju i na ulicę.

Obszar gry obejmuje kilka budynków i trzy ulice Brooklinu. Na ulicach można znaleźć skrzynki, śmietniki i budki telefoniczne można je przesuwać. UWAGA! nie jesteś w mieście sam. Morderców nie traktuj inaczej jak pięścią, a gdyby nie byli „rozmowni" strzelaj. Natkniesz się też na płatnych informatorów — zaoferuj im pieniądze za usługi, spróbuj dowiedzieć się czegoś o KASECIE. (Rozmowa — to krótki tekst wpisany w dymku SPEECH po naciśnięciu ENTER).

Chodząc we wskazanym rewirze możesz spotkać kobietę. Kobiety są tylko dwie, dwie siostry bliźniaczki, są identyczne, ale... jedna z nich dobrze Ci życzy, spróbuj konwersacji, ona zna drogę do twego celu i przy odpowiedniej argumentacji może Cię do niego zaprowadzić lub wskazać drogę. Podobnie płatni informatorzy po dłuższej konwersacji „puszczają parę" i mogą szepnąć coś ponadplanowo. Na mapie są oznakowane trzy przybliżone i jedno na 90% miejsca pobytu, jednej lub obu, sióstr, ale często ich „rozmieszczenie" jest losowe, tak jak losowe jest umiejscowienie sejfu z KASETĄ czy miejsce pobytu Don Prizzi'ego.

Kobieta prowadząca Cię do szefa jest narażona na wiele niebezpieczeństw, płatni mordercy mają za zadanie zlikwidować ją. Ty odpowiadasz głową za jej życie.

Co powinieneś posiadać, aby twoja misja zakończyła się sukcesem:

Pistolet — mały samopowtarzalny LILIPUT kal. 0,76 — jako rozstrzygający argument w kwestiach spornych;

Pieniądze — plik nowych banknotów do rozwiązywania języków niezdecydowanym;

Whiskey — prezent dla nowych znajomych, tym nie pogardzi nawet morderca;

Granat — bicz na opornych strażników, lub narzędzie rozrywki dla wroga;

Torba — przecież trzeba to wszystko w czymś trzymać.

(Gdybyś którejś z rzeczy nie mógł dosięgnąć, spróbuj zrzucić ją inną — UWAGA! nie radzę używać do tego celu granatu).

Niestety nie można podać konkretnej recepty na skończenie tej gry, ponieważ zbyt wiele rzeczy dzieje się tu losowo ale ogólnie, mając wszystkie rzeczy, należy odnaleźć przychylną tobie bliźniaczkę i namówić ją do współpracy, odnaleźć mafioza i zmusić go do mówienia, otworzyć wskazany przez niego sejf (odpowiednim szyfrem), zabrać KASETĘ i wrócić do biura.

I jeszcze parę wskazówek:

1. W twoim rewirze są dwie papugi — mogą one powtarzać zasłyszane gdzieś słowa, wykorzystaj je do swoich celów odpowiednim słowem... jakim musisz dojść do tego sam.
2. W niektórych drzwiach stoją strażnicy pilnujący pewnych tajemnic. Można ich usunąć przy użyciu przedmiotu lub odpowiednim hasłem. Hasło to może przekazać któryś z informatorów, więc rozmawiaj z nimi.
3. Zauważ, że psy, kule i strażnicy mają często podobny algorytm ruchu i umiejętnie je wykorzystaj, bo masz tylko jedno życie.
4. W trakcie sparingu z wrogiem nie musisz odwracać się do niego przodem, ponieważ zrobisz to za chwilę całkiem automatycznie.
5. Nie wchodź do żadnego z pomieszczeń środkiem drzwi, bo kulka murowana.
6. Kobieta, która każe ci „wykańczać" innych, nawet morderców, dobrze tobie nie życzy.

Opcje przy stracie (zmiana klawiszem C).

Rotate	— automatycznie omija przeszkody, skręca przed nimi;
Directional	— idzie (od razu) w nakazanym kierunku;
IQ on	— inteligencja włączona — przechodzi przez drzwi po oparciu się o framugę;
off	— inteligencja wyłączona — trzeba dokłanie trafić w drzwi.

Klawisze operujące w ZX Spectrum:

1 do **0**	— strzał, zmiana opcji
Q do **P**	— w górę
A do **Ent**	— w dół
Caps,X,V,N,S	— w lewo
Z,C,B,M,Space	— w prawo

Wesołej zabawy! Pokoi jest 199, za każdymi drzwiami coś nowego, ale uważaj bo masz tylko jedno życie.

(M1)

Bajtek

ULICA III

ULICA I

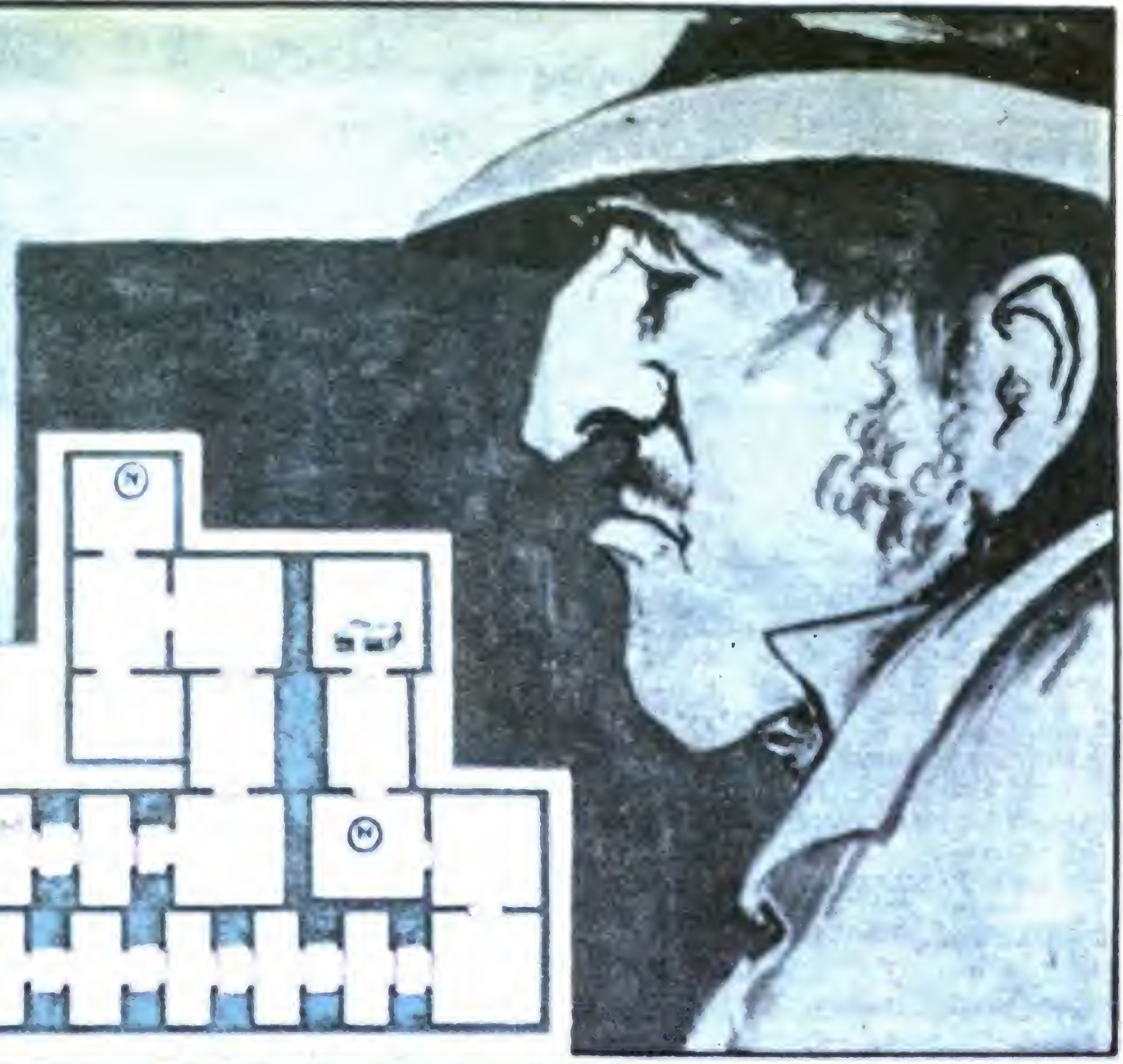
MOVIE



New York City









- MIEJSCE STARTU
- PISTOLET
- TORBA
- PIENIĄDZE
- GRANAT
- BUTELKA
- PSY
- KULE
- SKRZYNKI
- PAPUGA
- KOMPUTER

- DUŻE POMIESZCZENIE
- POKÓJ PRZEJŚCIOWY
- MAŁY POKÓJ
- ŚLEPY POKÓJ
- POM. PRZEJŚCIOWE

M – PŁATNI MORDERCY – WALCZ Z NIMI

I – PŁATNI INFORMATORZY – ROZMAWIAJ Z NIMI I PROPONUJ PIENIĄDZE

K – KOBIETY – ZAZWYCZAJ MOŻESZ JE SPOTKAĆ [illegible] LOSOWO

(N) NIERUCHOMY STRAŻNIK

(R) RUCHOMY STRAŻNIK

$(M)_1$ STRAŻNIK OTWIERA DRZWI NA HASŁO "PUZZLE"

$(M)_2$ STRAŻNIK PRZESUŃ GO ZA POMOCĄ PRZEDMIOTU

$(M)_3$ STRAŻNIK PRZESUŃ GO ZA POMOCĄ PRZEDMIOTU [illegible]

$(M)_4$ STRAŻNIK OTWIERA DRZWI NA HASŁO "OPEN DOOR"

$(M)_5$ STRAŻNIK [illegible]

[illegible]

LICZBA PRZED POSTACIĄ OZNACZA ILOŚĆ OSÓB W POMIESZCZENIU



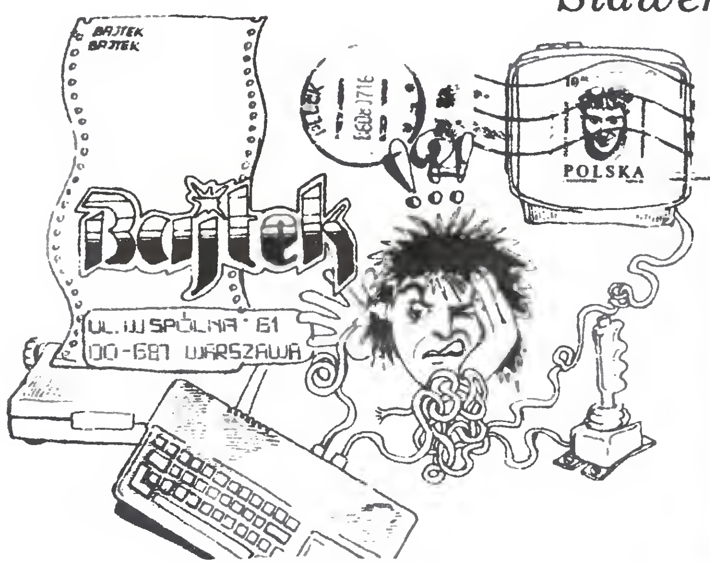
Już siedem miesięcy liczy sobie nasza Bajtkowa Lista Przebojów. Początkowo gościły na niej gry tylko na Spectrum. Obecnie pojawiły się już na Amstrada, Commodore i Atari.

Wielu czytelników zarzuca nam, że prezentowane na naszej liście gry nie są najnowsze, ale nie zapominajmy, że na listę oddają głosy czytelnicy z całej Polski. A teraz już statystyka: nadeszło 2538 propozycji, głosowano na 761 gier.

1	IMPOSSIBLE MISSION	
2	BOMB JACK	▲
3	SABOTEUR	▲
4	ENIGMA FORCE	!
5	MOVIE	!
6	BACK TO SKOOL	!
7	ROAD RACE	▼
8	YIE AR KUNG-FU	!
9	FAIRLIGHT	▼
10	ROLAND RAT RACE	▼

Nagrodę za najlepszy opis, zestaw programów do Commodore 64, otrzymuje Dominik Szweda z Gdańska. Programy ufundowała brytyjska firma ELECTRONICS EXPORT.

Sławek





KENNEDY APPROACH

Tym razem amatorom gier komputerowych przedstawiamy grę dość niezwykłą — symulację kontroli ruchu lotniczego. Ze względu na tematykę nie cieszy się ona tak wielką popularnością jak np. „HOBBIT" czy „THE WAY OF THE EXPLODING FIST" jednakże wzbudza co najmniej takie same, o ile nie większe, emocje jak dwie powyższe. Awaryjne lądowania, burze, separowanie samolotów, symulacja głosu ludzkiego i wiele, wiele innych atrakcji stawiają grę „KENNEDY APPROACH" w rzędzie tych najlepszych.

„DELTA 101 you are cleared to land" — DELTA 101 zezwalam lądować — te słowa wypowiada codziennie kilkadziesiąt razy kontroler ruchu lotniczego, człowiek od którego decyzji zależy bezpieczeństwo i los załóg oraz pasażerów wszystkich samolotów komunikacyjnych. Dzięki firmie MICROPROSE i Ty również możesz się stać takim człowiekiem wybranym spośród tysięcy. Twoim radarem będzie ekran telewizora, mikrofonem joystick. Po uruchomieniu gry stajesz się kontrolerem nie byle jakiego organu, bo kontroli zbliżania gdzie krzyżują się trasy wszystkich samolotów lądujących i startujących.

„Kennedy Approach" należy do jednej z tych nielicznych gier symulacyjnych, w których zamiast rzeki krwi kosmitów — najeźdźców, trzeba się nieźle wysilić by wszystkie samoloty dotarły bezpiecznie do celu. Swoje zdolności w tym kierunku możesz sprawdzić na pięciu lotniskach w Atlancie, Dallas, Denver, Waszyngtonie i Nowym Jorku. Pięć poziomów trudności jest tak pomyślanych, by grający po osiągnięciu wprawy na jednym z lotnisk mógł przenieść się na inne o podobnym natężeniu ruchu. Każda gra trwa (w zależności od poziomu) 10 lub 15 minut. Po upływie tego czasu komputer ocenia (bez taryfy ulgowej!) trafność podejmowanych przez Ciebie decyzji i w zależności od wyniku „nominuje" gracza na wyższe stanowisko. Oceniane są: spowodowane kolizje (w tym także niebezpieczne zbliżenia), opóźnienia w dolotach i startach, przesyłanie samolotów do innych organów (poza ekran) na odpowiedniej wysokości i w ściśle określonym punkcie.

Jak wiadomo, w lotnictwie nie brak również niespodzianek czy sytuacji nieoczekiwanych. Na wyższych poziomach można się spotkać z nagłą burzą, awaryjnym lądowaniem z powodu braku paliwa czy piekielnie wolnym „małym lotnictwem" — samolotami prywatnymi.

Bardzo dużym atutem tej gry jest wiernie symulowana mowa ludzka uzupełniająca i potwierdzająca wydane za pomocą joysticka rozkazy dla pilotów. Głos jest czysty i wyraźny, natomiast odpowiedzi pilotów („Roger") słychać na tle pracujących silników samolotu ... Grający może skierować samolot na jeden z ośmiu kursów (000°, 045°, 090°, 135°, 180°, 225°, 270°, 315°), ustawić samolot w strefie oczekiwania nad radiopomocą, spytać o aktualną wysokość i kurs oraz zniżać lub nakazać wchodzenie w zakresie 0 — 5000 stóp. Samoloty gotowe do startu kontroler wywołuje za pomocą klawiatury, w miarę rozwoju sytuacji.

Ekran został podzielony na trzy części. Pierwsza to skrótowy opis wszystkich samolotów (identyfikator, lotnisko startu i docelowe, wysokość). W części drugiej (pojedyncza linia) ukazują się wszystkie wydawane przez grającego polecenia). Pozostałą część ekranu zajmuje plan lotniska i aktualna sytuacja ruchowa. Jeżeli na ekranie nic się nie dzieje, to grę można przyspieszyć za pomocą klawisza spacji — powoduje to podwójny upływ czasu: klawisz F1 natomiast służy do zatrzymania gry w dowolnym miejscu.

Po zakończeniu gry, tak jak w prawdziwej kontroli lotów można jej przebieg zobaczyć powtórnie, a nawet zapisać go na dysku w postaci zbioru.

Przełożenie (udane!) zasad kontroli ruchu lotniczego na język komputerowy, różnorodność sytuacji, ciekawe opracowanie graficzne i dźwiękowe a także atrakcyjna tematyka stawiają, moim zdaniem, „Kennedy Approach" w czołówce gier symulacyjnych. Polecam tę grę tym wszystkim, którzy dosyć mają zabaw wojenno-kosmiczno-labiryntowych i chcieliby spróbować swych sił w całkiem nowym temacie — kontroli ruchu lotniczego.

(kd)

MASZ SZANSĘ

Te najnowsze gry nadesłane nam przez brytyjską firmę wysyłkową ELECTRONICS EXPORT z Londynu, zostaną rozlosowane wśród Czytelników, którzy nadeślą swoje propozycje do najbliższej Bajtkowej Listy Przebojów.





NONTERRAQUEOUS

Akcja tej gry toczy się na dalekiej planecie Nonterraqueous rządzonej przez diabolicznego tyrana, którym jest komputer. Używa on ludzi jako pionków do gry w kosmiczne szachy.

Dręczona ludzkość zbuntowała się i postanowiła zbudować robota, tzw. „poszukiwacza", którego zadaniem będzie odnalezienie bazy komputerowej ukrytej głęboko we wnętrzu gigantycznej góry i unicestwienie jej.

„Poszukiwacza" skonstruowano w wielkiej tajemnicy z elementów wykradzionych z linii montażowej fabryki robotów. Po wielu miesiącach sekretnych przygotowań „poszukiwacz" jest gotów do rozpoczęcia niebezpiecznej misji.

Teraz Ty musisz przeprowadzić „poszukiwacza" przez wnętrze góry i zniszczyć bazę komputerową. Musisz przebyć 42 poziomy, stanowiące w sumie trzy połączone ze sobą sekcje. Najwyżej znajduje się sekcja jaskini, niżej — sekcja maszynerii głównej, na dnie — sekcja podzespołów. Znajduje się tam w sumie około 1000 pomieszczeń. Poszukiwanie bazy i poruszanie się naprzód nie będzie więc łatwe!

Zetknięcie się z jakimkolwiek z pilnujących komputerowego władcę sług zmniejsza twoją energię życiową a w niektórych wypadkach kończy się fatalnie. Te i wiele innych niebezpieczeństw powinieneś ominąć i śmiało dążyć do celu. Dotrzesz tam jednak tylko wtedy, gdy użyjesz całego swojego intelektu, a jak trzeba to i... kurka spustowego. Przykład: na dnie szybu centralnego znajduje się rakieta, z której nie będziesz mógł jednak skorzystać jeśli wcześniej nie odnajdziesz paliwa. A tylko za pomocą rakiety możesz bezpiecznie przedostać się przez komorę gazową...

Twój „poszukiwacz" może przybierać dwa kształty, a co za tym idzie i dwa sposoby działania: sferyczny, w którym możesz używać miotacza promieni i sposób pozbawiony obrony, w którym tego ognia używać nie możesz. Transformacji dokonać można w sali z przedmiotem pod nazwą SWOP.

Jeżeli „poszukiwacz" zostanie bardzo osłabiony atakami fotonów — ratunkiem może stać się dodatkowa porcja energii, którą możesz pobrać w kilku salach z pojedynczo stojącymi pojemnikami.

Poszukiwaczem kierujesz przy pomocy drążka sterowego lub klawiatury:

Q — ruch do góry　　P — ruch w prawo
A — ruch w dół　　O — ruch w lewo
SPACE — ogień
I — nabycie przedmiotu, uzyskanie energii, wybór kształtu
U — zrzucenie bomby,

Gra jest bardzo ciekawa i... bardzo trudna. Nie upieraj się więc aby dojść do celu od pierwszego czy drugiego dnia!

Podpowiemy Ci jeszcze, że nie ma lepszej metody na przebycie zapory elektrycznej niż wysadzenie jej w powietrze, a w przeszkodzie stałej można przebić sobie przejście miotaczem promieni. Ale dalej radź już sobie sam. Pamiętaj: udręczeni mieszkańcy Nonterraqueous czekają na Twoją pomoc!

(ps)



CHUCKIE EGG

CHUCKIE EGG to gra znana posiadaczom wielu typów komputerów m.in. ZX SPECTRUM i AMSTRAD

Obraz na ekranie składa się z bardzo prostych elementów: drabin i platform. Tworzą one kurnik pewnego farmera, który ma kłopoty w podbieraniu kurom jajek. KURY SĄ AGRESYWNE!!! Zderzenie z ptakiem źle kończy się dla człowieka...

Twoim zadaniem jest poprowadzenie farmera z platformy na platformę i zebranie wszystkich jajek, zanim kury zjedzą ziarno, rozmieszczone kupkami w różnych miejscach kurnika. Wędrówka pomiędzy platformami możliwa jest po drabinach, łączących poziomy lub z pomocą krążących wind (w dalszych pomieszczeniach kurnika). Kupki ziarna także mogą być zbierane i przyniosą ci punkty, ale mogą też być przyczyną przedwczesnego zakończenia gry, gdy ostatnia kupka ziarna zostanie zjedzona przez kury lub zabrana przez ciebie.

Sterowanie farmerem jest utrudnione... Wchodzenie po drabinie jest możliwe tylko wtedy, gdy staniesz dokładnie pod nią. Jeśli opanujesz ustawianie się nad lub pod drabiną to inne ruchy nie będą sprawiały ci trudności. Bieganie w górę i w dół jest bardzo proste, skoki też. Szczególnie trudne jest wskakiwanie i wyskakiwanie z windy. Natomiast spadanie z platformy na platformę nie jest groźne i niejednokrotnie może uratować Ci życie.

Klucze sterujące to: A — góra, Z — dół, , (przecinek) — lewo, . (kropka) — prawo, SPACJA — skok, S — uruchamia grę na początku, H — zatrzymanie w trakcie gry, ESC + H powrót do menu. Po wybraniu K na początku, gdy na ekranie jest menu, można przedefiniować klucze wg własnych upodobań. Drążek sterowy nie działa. Po wybraniu S pojawia się na ekranie pytanie o ilość graczy. Wystarczy nacisnąć 1 — 4, by móc zagrać samemu lub z trzema kolegami. Gracze przechodzą kolejno pomieszczenia. Następny rozpoczyna kolejkę po „wpadce" poprzednika.

Komunikaty u góry ekranu: wynik (sześć cyfr), pod nim kreskami oznaczona ilość „zapasowych" farmerów. Poniżej od lewej: ilość graczy, poziom (w trybie demonstracyjnym ukazuje się tu nawet liczba 30?), premia (startuje od 1000 i zmniejsza się wraz z upływem czasu) i wreszcie czas, który startuje od 900.

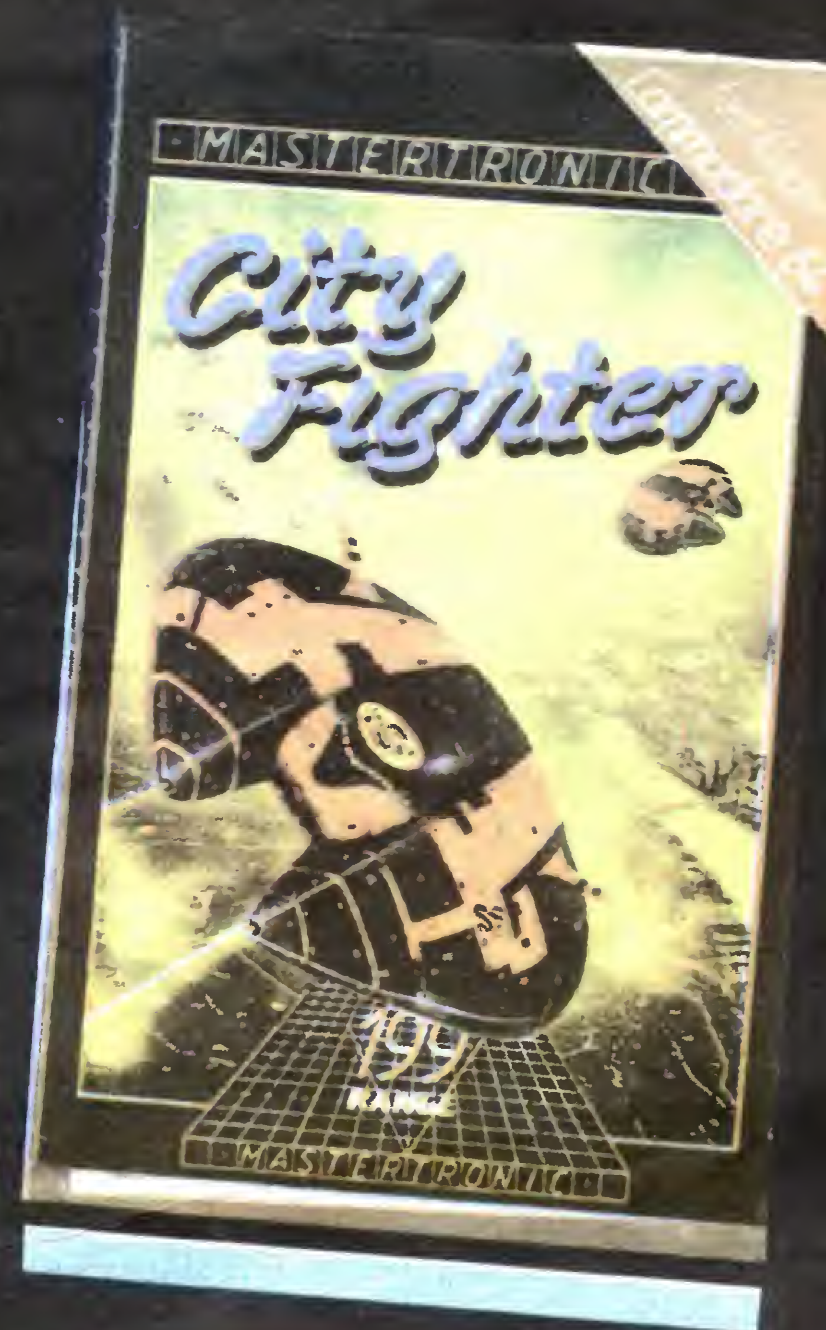
Bardzo trudno jest przejść więcej niż 5 pomieszczeń kurnika, dlatego nie możemy podać zbyt szczegółowych informacji o wszystkich jego zakamarkach.

Grafika nie jest rewelacyjna, wszystkie pomieszczenia są w tych samych kolorach. Gra zajmuje tylko 9K pamięci i podana w trybie demonstracyjnym ilość 30 pomieszczeń wydaje się wątpliwa...

(tp)







POLSKI ALFABET cz. I

Dla przekonania tych, którzy twierdzą, że bez polskiej pisowni można się obejść, zamieszczam przykład zaproponowany przez prof. W. M. Turskiego: „ZADANIE KATA NA LACE". Sposobów interpretacji tego zdania jest na tyle dużo, aby straciło ono sens w ogóle.

Najprostszy sposób uzyskania typowo polskich liter polega na uzupełnieniu tekstu przecinkami, kropkami itp. tak, aby powstałe w ten sposób znaki zastępowały nasze ó, ź, ą czy ń. Inny sposób, wiele trudniejszy, ale pozwalający na przystosowanie do pracy w języku polskim istniejących programów, polega na przedefiniowaniu matrycy (generatora) znaków, zapisanego w pamięci komputera. Sposób ten zostanie szczegółowo opisany w kolejnych odcinkach cyklu.

Na początek jedna z wymienionych metod.
Możemy uzupełniania liter na ekranie, dokonywać „ręcznie" podczas pisania programu, lub też przystosować system operacyjny do drukowania kilku znaków zamiast jednego tak, by w efekcie otrzymać polski znak. Druga metoda wymaga użycia języka maszynowego, na razie więc zajmiemy się pierwszą. Jedynym praktycznym zastosowaniem tej metody jest drukowanie na ekranie opisów lub informacji dla użytkownika, np. „naciśnij dowolny klawisz, aby kontynuować". Możemy wybrać jedną z trzech możliwości. Pierwsza z nich, to drukowanie na przemian linii liter i linii „uzupełnień", np.

```
5 PRINT"            ,         ,"
10 PRINT"do najdonioślejszych
wynalazków"
15 PRINT"                     ,"
20 PRINT"w przemyśle
mikroelektronicznym"
25 PRINT"         ."
30 PRINT"należy
mikroprocesor–dzie£o..."
```

Jak widać, jest to straszna rozrzutność. Większość miejsca w liniach o nieparzystej numeracji zajęta jest przez spacje, czyli niewykorzystana. Aby temu zaradzić, można zastosować drugą możliwość, polegającą na wstawianiu owych uzupełnień tylko tam, gdzie jest to konieczne. W praktyce wygląda to następująco:

5 PRINT"do najdonios↑←↓lejszych wynalazko↑←,↓"

gdzie strzałki oznaczają odpowiednie znaki kontrolne ruchu kursora (czyli dziwne znaczki, dobrze znane użytkownikom Commodore). Podczas drukowania takiej jak powyższa linii komputer sam stawia przecinek nad s i wraca do drukowanej linii. Dodatkową zaletą takiego „upakowania" tekstu jest możliwość swobodnego nim operowania, tzn. umieszczenia w tabelach, wysyłania na dysk czy pozycjonowania na ekranie. Aby uzyskać litery takie jak ą czy ę, wymagające dostawienia przecinka z dołu, należy po prostu zamienić miejscami pierwszy i trzeci znak kontrolny (strzałki w górę i w dół). Literę Ł uzyskać można, używając znaku funta (sposób organizacji ekranu tekstowego w C 64 uniemożliwia postawienie dwóch znaków w jednym miejscu, a więc nie można po prostu przekreślić L).

Trzecia możliwość wymaga użycia grafiki wysokiej rozdzielczości. Jej zastosowanie ma sens tylko w przypadku użycia któregoś z dostępnych rozszerzeń BASIC-a, jak np. SIMON'S BASIC. Idea jest taka sama jak poprzednio, z tym, że używając grafiki nie jesteśmy ograniczeni organizacją ekranu i „uzupełnienia" można postawić tak, by uzyskać jak najbardziej czytelny efekt. Niestety, plusy tej metody ograniczają się do estetyki, bo ilość pracy włożonej w uzyskanie dobrych efektów jest o wiele większa niż poprzednio.

Nie podaję przykładów, ponieważ komendy graficzne nie są znormalizowane i każde rozszerzenie BASIC-a wymaga nieco innego podejścia. Ogólnie rzecz biorąc, po wypisaniu na ekranie linii tekstu, należy go uzupełnić za pomocą komend takich jak PLOT, LINE czy DRAW o odpowiednio dobranych parametrach.

(ms)



CZYSZCZENIE KLAWIATURY COMMODORE 64

Długie i intensywne używanie komputera powoduje po pewnym czasie użytkowania zanieczyszczenie klawiatury. Objawia się to poprzez podwójne wyświetlanie znaku wciśniętego klawisza, brak kontaktu itp. W Commodore 64 zanieczyszczenie to powoduje przede wszystkim wadliwe działanie klawisza RETURN oraz klawiszy cyfrowych. Na szczęście jest na to prosta rada.

Potrzebne nam będą następujące narzędzia i śrubokręt krzyżowy, lutownica oraz kawałek suchej flanelowej szmatki.

Po wykręceniu trzech śrub łączących obie połówki obudowy należy odłączyć klawiaturę od płytki. Następnie należy wykręcić 6 śrub mocujących klawiaturę do obudowy oraz odlutować dwa druciki łączące klawisz SHIFT LOCK z płytką.

Zwracając baczną uwagę na przewody wyprowadzone z płyty wykręcamy wszystkie śrubki usztywniające płytkę. Proponujemy złożyć je do jakiegoś pojemniczka gdyż z własnego doświadczenia wiem jak długo można tych śrubek potem szukać. Po ich wykręceniu płytka daje się swobodnie oddzielić od reszty klawiatury i jest gotowa do czyszczenia.

Głównym składnikiem zanieczyszczeń jest plastikowy pył powstający na skutek ścierania się prowadnicy i elementu prowadzącego w trakcie wciskania klawisza. Pył ten daje bardzo łatwo się usunąć za pomocą suchej szmatki (nie jest konieczne przemywanie spirytusem, przestrzegamy natomiast przed stosowaniem spirytusu salicylowego gdyż pozostawia on po sobie duży osad). Za pomocą pędzelka należy również usunąć nagromadzony pył z prowadnicy oraz elementów zwierających wszystkich klawiszy. Po tak przeprowadzonej konserwacji można komputer zmontować ponownie — klawiatura będzie jak nowa.

(kd)

TAJEMNICE C — 128

Po włączeniu, Commodore PC-128 rozpoczyna pracę w trybie 128. Do trybu C-64 możemy przejść za pomocą rozkazu GO64. Jest też drugi sposób, dużo szybszy i wygodniejszy. Wystarczy podczas włączania lub wykonywania „reset" trzymać wciśnięty klawisz Commodore. Jeżeli podczas pracy w trybie C-64 zrobimy reset — komputer wróci do trybu 128. Można tego uniknąć, wprowadzając następujące instrukcje:

```
BANK 1 : POKE 65528,77 : POKE
65529,255 : GO 64
```

Od tego momentu komputer zawsze po resecie będzie wracał do trybu C-64.

Jeżeli podczas pracy na PC-128 skasujemy instrukcję NEW program w BASICU-u, można go łatwo odtworzyć:

```
POKE PEEK(45)+256*PEEK(46)+1,28:SYS DEC("4F4F")
```

Po wykonaniu "reset" PC-128, program w BASIC-u zostaje skasowany. Jeżeli jednak naciskając przycisk "reset" będziemy równocześnie trzymać wciśnięty klawisz RUN/STOP. PC-128 przejdzie do trybu MONITOR'a. Trzeba z niego wyjść przez naciśnięcie X i RETURN. Teraz możemy kontynuować pisanie naszego programu — nie został skasowany.

Przemysław Koziarski



Od chwili kiedy został skonstruowany Commodore 64 można zaobserwować nieustającą lawinę książek dotyczących programowania tego komputera. Książka o której mowa dotyczy programowania w języku maszynowym, jednakże jej charakter jest nieco odmienny od pozostałych publikacji tego rodzaju. Zawiera ona 8 rozdziałów i 9 załączników w których omówiono (podobnie jak w większości takich książek) rodzaje zapisu, instrukcje mikroprocesora, tryby adresowania itp. Cały opisany materiał wydaje się jednak być przeznaczony dla Czytelnika posiadającego już pewne wiadomości na temat programowania w asemblerze, gdyż zbyt mało jest konkretnych przykładów zastąpionych tu schematami blokowymi. Ilustrują one działanie poszczególnych systemów i układów komputera co na pewno jest pomocne, ale chyba nie tego oczekuje użytkownik, stawiający pierwsze kroki w tej dziedzinie.

„Beyond BASIC on your Commodore 64" ma także kilka (i to mocnych) plusów. Pierwszy z nich to trzy stosunkowo długie programy dotyczące grafiki — zmiana rodzaju liter, grafika wysokiej rozdzielczości (HIRES) oraz sprite'y. W ogóle grafika jest mocno w tej książce faworyzowana — jest to pomocne dla wszystkich tych, którzy myślą o układaniu własnych gier komputerowych w języku maszynowym. Jako drugi plus należy zaliczyć dokładny opis pamięci od adresu 0 do 1023. Trzecim strzałem w dziesiątkę jest dwustronicowe omówienie sposobów zabezpieczenia własnych programów wraz z przykładem umożliwiającym tworzenie własnych programów samouruchamiających się.

Ze względu na dość ogólne choć niewątpliwie potrzebne informacje oraz brak przykładów (za wyjątkiem grafiki) polecałbym tę książkę programistom mającym już pewne doświadczenie w programowaniu w języku wewnętrznym mikroprocesora MOS 6510 oraz wszystkim miłośnikom grafiki komputerowej.

Klaudiusz Dybowski

C.I. Burkinshaw
„Beyond BASIC on your Commodore 64"
Sigma Press, Wielka Brytania, 1984
ISBN 0-905-104-91-9 Stron 144

Od zarania dziejów historia ludzkości jest historią wojen. Wynika to niestety z niedoskonałości ludzkiej natury. Na szczęście nie wszyscy zamiłowani w sztuce wojennej od razu rozpętują wojny. Są jeszcze gry wojenne. Od najprostszych, wręcz prymitywnych do skomplikowanych gier — symulacji, pozwalających na stosowania przemyślnych strategii i wybiegów taktycznych. Co prawda nie pochwalam zainteresowania sztuką wojenną — zbyt wielu ludzi cierpiało i cierpi z tego powodu — jednak przyznaję, że ciekawym przeżyciem jest postawienie się w roli Napoleona czy Aleksandra Macedońskiego w przededniu jednej z ich wielkich bitew. Jednak jeżeli ktoś chce zabawić się w prowadzenie gry wojennej, pojawia się mnóstwo niespodziewanych przeszkód. Potrzebna jest przecież plansza, na której rozgrywać będziemy bitwę. Potrzebne są figurki pojazdów i żołnierzy. Skąd to wszystko wziąć? Jest prostszy sposób. Cały nieodzowny „wystrój" można przecież zaprogramować na ekranie komputera. Czy tylko to? Komputer można obarczyć również pamiętaniem zbioru reguł, stanu uzbrojenia armii, sytuacji na planszy itp. Krótko mówiąc, swój udział w prowadzeniu gry można ograniczyć do czysto umysłowego. O tym, jak taki stan rzeczy osiągnąć, traktuje książka pt. „Commodore 64 Wargaming", czyli „Prowadzenie gier wojennych na Commodore 64". Pozycja ta przeznaczona jest przede wszystkim dla dwojakiego rodzaju czytelników: dla miłośników gier wojennych, którzy nie mogą dla siebie znaleźć nic interesującego wśród istniejących programów, oraz dla programistów, którzy poszukują nowej, interesującej dziedziny.

W kolejnych rozdziałach czytamy o teorii gier wojennych, o sposobach umieszczania w pamięci map, danych o uzbrojeniu i pozycji armii, o metodach rozwiązywania różnych problemów związanych z przeniesieniem akcji do wnętrza komputera, wreszcie na zakończenie autorzy proponują kilka gotowych programów, które mogą zostać wykorzystane zarówno jako gotowe gry, jak i jako wzór przy pisaniu własnych programów. Na uwagę zasługuje tu podejście autorów książki do organizacji struktury programów. Otóż każda gra wojenna składa się praktycznie z tych samych procedur, obsługujących ekran, aktualizujących tabele danych czy umożliwiających wykonywanie ruchów. Procedury te zostały osobno opisane i znormalizowane tak, że podczas pisania własnego programu wystarczy tylko do zestawu gotowych procedur dodać kilkanaście własnych linii, zawierających główny algorytm gry. Podobna sytuacja ma miejsce w przypadku tabel z danymi. Potraktowane są one jako osobne zbiory na taśmie lub dysku, dzięki czemu można do istniejących danych w prosty sposób wprowadzać poprawki, lub też całkowicie je wymieniać, np. zmieniając terytorium działań w swej ulubionej grze. Sposób zapisu danych jest znormalizowany, co pozwala na wymienianie danych między różnymi programami. W dodatkach na końcu książki opisane zostało zastosowanie wszystkich użytych zmiennych, zresztą identycznych w każdym programie.

Podsumowując, można stwierdzić, iż książka, mimo że wymaga od czytelnika posiadania pewnych podstawowych przynajmniej wiadomości, godna jest uwagi, a systematyczność autorów — warta naśladownictwa.

Michał Silski

Owen Bishop Audrey Bishop
„Commodore 64 Wargaming"
Collins Professional and
Technical Books Londyn





Tym razem mam przyjemność zaprezentować bardzo interesującą książkę poświęconą programowaniu w języku maszynowym komputera Commodore 64.Jest ona przeznaczona dla studentów i Czytelników dobrze znających BASIC (lub PASCAL, FORTRAN czy PL(I)), oraz nieźle zorientowanych w programowaniu w asemblerze. W stu sekcjach, z których składa się książka, zawarto dokładnie omówione poszczególne zagadnienia wraz z trzema pytaniami kontrolnymi na końcu każdej z nich. Dodatek stanowi 17 bardzo pomocnych w programowaniu tablic.

Do wszystkich programów opisanych w książce autor używał popularnego asemblera DEVELOP 64, również dokładnie w książce opisanego. W razie jego braku można korzystać z innego (np. MACRO-ASSEMBLER), jednakże wszystkie programy były układane za pomocą wyżej wymienionego. Jest to, moim zdaniem, dodatkowy plus tej publikacji, gdyż w ten sposób książka stanowi doskonałe uzupełnienie dla wszystkich posiadaczy DEVELOP 64.

Ujęcie problematyki programowania w kodzie wewnętrznym mikroprocesora 6510 z matematycznego punktu widzenia sprawia, że książka ta jest jedną z bardziej wartościowych pozycji dotyczących Commodore 64. Pozwala ona bowiem nie tylko na bardzo dobre opanowanie techniki programowania, ale jest także skarbnicą wiedzy niezbędnej każdemu programiście z prawdziwego zdarzenia. Polecam tę książkę wszystkim tym, którzy zapoznali się już z podstawami i myślą o poważnym kontynuowaniu i rozwijaniu zainteresowań językiem maszynowym Commodore 64.

(kd)

W. Douglas Maurer
„Commodore 64 Assembly Language"
Computer Science Press, USA, 1985
ISBN 0-88175-040-9
Stron 416

PORADNIK MŁODEGO PIRATA cz. III



Z poprzedniej części dowiedziałeś się drogi Czytelniku dlaczego przy niektórych długich programach Twój komputer odmawiał Ci posłuszeństwa. Jak już wspominaliśmy także i program „TURBOTAPE 64" nie był pozbawiony tej wady i programy dłuższe niż 38911 bajtów traktował po macoszemu odmawiając ich zapisania na taśmie. Czy jest na to jakaś rada? Oczywiście.

Rozwiązaniem Twoich kłopotów jest jedna z wielu wersji programu **TURBOTAPE — SUPERTURBO** z tytułem (wyświetlanym po uruchomieniu programu) **„THE MASTER PRESENTS SUPERTURBO"** W dalszej części naszego „Poradnika" program ten będziemy nazywać po prostu **MASTER**. Nie wdając się zbytnio w szczegóły techniczne pozwala on na zapisanie CAŁEGO obszaru pamięci od adresu 2049 ($ 0801) do 53247 ($ CFFF) włącznie, a w niektórych wypadkach nawet trochę więcej. Za pomocą tego programu zapisywaliśmy już programy dyskowe mające w katalogu (directory) 207 bloków!!!. Zwykle **TURBOTAPE** nie ma więc do **MASTER-a** żadnego porównania, gdyż kłopoty z zapisem mogą wystąpić już z programami o długości 155 bloków. **MASTER** jest w pełni kompatybilny ze zwykłym **TURBO**, tak więc wystarczy tylko zamienić by kłopoty z zapisem długich programów zniknęły bezpowrotnie. Ma on niestety jedną wadę — nie daje się ponownie uruchomić za pomocą **SYS 50000**.

Wszystkich zainteresowanych uzyskaniem tego programu informujemy, że będzie można go dostać w naszym klubie komputerowym **„MANIAK", Warszawa-Ursynów ul. Wasilkowskiego 7 (tel. 40-62-64)** jednakże dopiero po rozpoczęciu kursów co nastąpi 16 października 1986 r.

Na czym polega fenomen? Otóż **MASTER** działa generalnie rzecz biorąc z dwóch miejsc pamięci. To pierwsze to podobnie jak przy zwykłym **TURBO** — dodatkowe 4 K pamięci od adresu $ C000 — s CFFF. To drugie to... obszar pamięci **RAM** systemem operacyjnym. Pamiętasz nasze rozważania na temat włączania i wyłączania interpretera BASIC? Jeżeli tak, to wiesz już także co robi **MASTER** i dlaczego pozwala na zapisanie prawie 52 K pamięci.

Gdyby nasze kłopoty z przegrywaniem programów dyskowych na taśmę ograniczały się jedynie do ilości bloków i stosowania odpowiednich dopalaczy, to ich przegrywanie byłoby po prostu fraszką. W rzeczywistości są jeszcze inne programy których ominąć czasami nie sposób. Zanim jednak o nich pomówimy wróćmy do różnicy pomiędzy **LOAD „NAZWA", 1 i LOAD „NAZWA", 1,1** (**←L „NAZWA", 1** i **←L „NAZWA", 1,1**). Co sygnalizuje komputerowi ta druga jedynka?

Pierwsza z nich to numer urządzenia, z którego będziemy program wczytywać (1 dla taśmy 8 dla dysku). To jest jasne i nie wymaga omawiania. A co robi ta druga?

Każdy zapisywany na taśmie czy dysku program jest zaopatrywany przez komputer w swoistego rodzaju wizytówkę zwaną potocznie nagłówkiem programu. Jak już się domyślasz zawiera on na pewno jego tytuł oraz bardzo ważne informacje dotyczące adresów pamięci, gdzie ma być on w późniejszym czasie wczytywany, rodzaju nagłówka itp. O ile program taśmowy może się bez tytułu obejść (jest to w końcu tylko udogodnienie dla użytkownika) o tyle program z uszkodzonym nagłówkiem (np. poprzez zgięcie taśmy) żadną miarą do pamięci wczytać się nie da. Warto też wspomnieć, że pomiędzy programem dyskowym i taśmowym są dość znaczne różnice. Program taśmowy ma nagłówek rozszerzony o jedną drobną, ale o kapitalnym wprost znaczeniu informację: znacznik relokowalności programu. Za program relokowalny uważa się taki program, który zaczyna się od adresu 2049 i był zapisany za pomocą **SAVE „NAZWA"**, 1 (lub po prostu **SAVE „NAZWA"** gdyż w wypadku braku numeru urządzenia system operacyjny automatycznie przyjmuje, że chodzi o zapis na taśmie). Za program nierelokowalny uważa się program zapisany za pomocą **SAVE „NAZWA", 1,1**. W zależności od ustawienia tego znacznika mamy dwie możliwości wczytania programu do pamięci: albo od adresu 2049, albo w obszar pamięci określony ściśle adresami zawartymi w nagłówku programu. Spróbujmy to teraz sprecyzować.

Z chwilą dokonywania zapisu system operacyjny komputera sprawdza wartości w komórkach 43, 44, 45, 46 i wpisuje je do nagłówka programu wraz z odpowiednio ustawionym znacznikiem relokowalności, zależnym do użytej formy **SAVE**. W czasie wczytywania znacznik ten jest przez komputer sprawdzany i w zależności od niego program zostaje wczytany albo od adresu 2049 albo zgodnie z adresami pamięci zapisanymi w nagłówku.

LOAD „NAZWA", 1 oznacza mniej więcej następującą procedurę: „załaduj do pamięci program rozpoczynając od adresu 2049 i ignorując adresy zawarte w nagłówku". **LOAD „NAZWA", 1,1** oznacza: „załaduj do pamięci program ZGODNIE z adresami zawartymi w nagłówku". Tu trzeba zaznaczyć, że w wypadku programu taśmowego zapisanego za pomocą **SAVE „NAZWA", 1,1** znacznik relokowalności spowoduje ZAWSZE wczytanie programu w obszar pamięci określony adresami zawartymi w nagłówku nawet jeżeli użyjesz do jego wczytania formy **LOAD „NAZWA"**. Ponieważ sposób wczytywania programu z dyskietki jest odmienny posłużmy się przykładem. Wyobraźmy sobie, że zapisujemy najpierw na taśmie, a następnie na dysku program zajmujący obszar pamięci od adresu 2049 do 14300. W obu wypadkach do nagłówka programu zostanie wpisana informacja, że program znajdował się w obszarze pamięci 2049-14300, bez względu na to czy użyjemy formy **SAVE „NAZWA", 1** czy **SAVE „NAZWA", 1,1**. W wypadku programu taśmowego zostanie również zapisany znacznik relokowalności w zależności od użytej formy rozkazu **SAVE**. Późniejsze wczytywanie do pamięci nie spowoduje żadnych różnic pomiędzy ładowanie z taśmy czy dysku gdyż program ten załaduje się w każdym wypadku od adresu 2049. Zgoła inaczej wygląda sprawa, gdy program który chcemy zapisać znajduje się np. pomiędzy adresami 12000–21555. Żeby zapisać tylko ten obszar pamięci konieczne jest uprzednie ustawienie adresu początku programu w komórkach 43, 44:

PRINT INT (12000/256), 12000 — INT (1200/256) *256
46 224
READY.
POKE 43, 224 : POKE 44, 46

Użycie formy **SAVE „NAZWA"** spowoduje zapisanie pamięci od adresu 2049 przez co niepotrzebnie zapiszemy 10 K zbędnej nam informacji. Musimy zatem użyć formy **SAVE „NAZWA", 1,1**. W nagłówku zostaną zakodowane adresy 1200-21555, a znacznik relokowalności zostanie ustawiony w pozycji „nierelokowalny" co miałoby znaczenie jedynie w wypadku programu taśmowego. Przeanalizujemy teraz w jaki sposób tak zapisany program będzie następnie wczytywany.

1. Program taśmowy:
— **LOAD „NAZWA"** spowoduje wczytanie programu pod adresy zawarte w nagłówku gdyż zadziała tu ustawienie znacznika relokowalności w pozycję „nierelokowalny";
— **LOAD „NAZWA", 1,1** spowoduje wczytanie programu także pod sprecyzowane w nagłówku adresy.

2. Program dyskowy:
— **LOAD „NAZWA", 8** spowoduje ze względu na brak znacznika relokowalności w nagłówku programu dyskowego wczytanie takiego programu od adresu 2049, a więc adresy w nagłówku rzeczywiście zostaną zignorowane;
— **LOAD „NAZWA", 8, 1** zadziała tak samo jak w wypadku programu taśmowego.

Ten mały trick z adresami ma dla nas duże znaczenie. Został wymyślony dla wygody programisty. Załóżmy, że ułożyłeś krótki program muzyczny lecz dane zawarte w instrukcji **DATA** zajmują dajmy na to 20 K. Ile czasu trzeba by było czekać, zanim Twój program zacznie działać! Znacznie wygodniejsze jest podzielenie procesu wczytywania na dwa etapy: pierwszy to wczytanie programu głównego — drugi to wczytanie danych dla procesora muzycznego w wybrany obszar pamięci... Innymi słowy: po co robić dużo instrukcji **DATA** gdy wystarczy tylko zapisanie obszaru pamięci gdzie się one znajdują.

W podanym wcześniej przykładzie ograniczyliśmy się jedynie do określenia adresu początku programu; należy więc pamiętać przy zapisywaniu danego obszaru aby został także odpowiednio określony jego koniec w komórkach 45 i 46; adres ten musi być zawsze zwiększony o jeden. Jeżeli np. program kończy się w komórce 21555 to w adresach 45 i 46 należy ustawić 21556. Jest to spowodowane tym, że komórki te informują system operacyjny o pierwszym wolnym bajcie pamięci PO zakończeniu programu.

(kd)

Klan Commodore redagują: Klaudiusz Dybowski i Michał Silski

KLAN NIETYPOWYCH

SHARP

Komputer ten, wraz z wbudowanym magnetofonem i drukarką, można kupić na giełdach za 170 tys. zł. W RFN jest on dostępny w takim zestawieniu w cenie poniżej 400 DM. Przyczyną tak niskiej ceny jest konkurencyjność cenowa wyrobów japońskich w stosunku do ich odpowiedników produkcji innych krajów, powodowana próbą opanowania rynku. Należy więc zaznaczyć, iż jest to najtańszy komputer w takiej konfiguracji dostępny na naszym rynku.

Możliwości **SHARP**-a **MZ-700** są o wiele większe od wielu innych komputerów popularnych w naszym kraju. Muszę jednak zastrzec, iż pisząc o możliwościach nie mam na myśli zabawy. W tym względzie Commodore 64 jest bezkonkurencyjny. Na początek proponuję zapoznanie się z kilkoma ważniejszymi danymi technicznymi:

Mikroprocesor **Z-80A** z zegarem **3,5 MHz.**

74 K pamięci z przełączalnymi bankami w tym:

64 K wolnej pamięci **RAM**

4 K V-RAM	pamięć ekranu z możliwością scrollingu w górę i w dół
4 K ROM	z systemem operacyjnym MONITOR posiadającym kilkanaście instrukcji i funkcji (podobny do Monitora w Commodore 128)
2 K	generator znaków z gotowymi 750 znakami: 140 znaków literowych dużych i małych alfabety angielski i niemiecki 610 znaków graficznych.

255 portów I-0 z wbudowanym interfejsem.

Profesjonalna i wygodna w obsłudze klawiatura **QWERTY** z 69 klawiszami i wydzielonymi blokami klawiszy kursora i klawiszy programowalnych z możliwością jednoczesnego zaprogramowania do 10 funkcji.

Wbudowany (wymienny) magnetofon cyfrowy zapewniający bezproblemowe wgrywanie programów z taśmy magnetycznej (typowe kasety).

Uwagę zwraca b.solidne i dokładne wykonanie całego komputera.

Model **MZ-700** nie posiada grafiki **High-Resolution** na ekranie, lecz brak ten rekompensują duże możliwości dodawanej razem z komputerem wewnętrznej 4-kolorowej drukarki typu Printer-Plotter, drukującej lub kreślącej na normalnym papierze szerokości 11.5 cm. Na uwagę zasługuje precyzja tego, wymiennego zresztą, elementu. Rozdzielczość plottera wynosi 480 pkt. na 2000 pkt.!

Korzystanie z drukarki jest możliwe tak za pomocą **BASIC-a** jak i **MONITOR-a ROM.** Można ją też używać jak maszyny do pisania — nie posiadając wogóle podłączonego telewizora czy monitora T.V.

Zestaw 750 gotowych znaków pozwala na szybkie i łatwe tworzenie na ekranie najbardziej skomplikowanych rysunków, napisów, tekstów itd. Dzięki jednemu wspólnemu trybowi pracy dla grafiki i tekstów, można używać na ekranie bez żadnych przełączeń 25 linii po 40 znaków w wierszu. Do dyspozycji jest 8 kolorów jednocześnie.

Wbudowany wewnątrz komputera wzmacniacz i głośnik wewnętrzny pozwalają na bardzo dobry odbiór dźwięku bezpośrednio z komputera, bez potrzeby posiadania telewizora w systemie **PAL.**

Malkontentów narzekających na jednokanałowy generator dźwięku lub brak grafiki **H-R** na ekranie, zadowoli model **MZ-800** o podobnym wyglądzie i podobnej cenie, z grafiką 640 pkt. na 200 pkt., 4-kanałowym dźwiękiem i 16 kolorami. Występuje przy tym całkowita wymienność programów z **MZ-700** na **800** (zarówno w **BASIC-u** jak i kodzie maszynowym. Oddzielne gniazda pozwalają na podłączenie monitora monochromatycznego kolorowego i zwykłego telewizora domowego. Przez dodatkowe gniazdo **RGB** wyprowadzone są ponadto wszystkie sygnały wizyjne.

Do komputera można podłączyć normalną drukarkę i dysk elastyczny. Wewnętrzny magnetofon można zastąpić wkładaną na jego miejsce stacją Quick-Disc 3.5". Można też podłączyć dowolny magnetofon zewnętrzny.

Istotną cechą komputerów **SHARP MZ-700** i **800** jest, iż programy są ładowane do zupełnie wolnej przestrzeni pamięci **RAM 64 K.**

System MONITOR rezydujący na stałe w jednym z banków w pamięci EPROM można następnie programowo odłączać, a banki pamięci dowolnie przełączać. W pamięci można jednocześnie umieszczać tyle programów, ile się zmieści.

Załączony przez firmę **SHARP** na oddzielnej kasecie Interpreter języka **BASIC** zajmuje 28 K RAM. Na uwagę zasługuje bardzo dobry edytor. W te-

Należy przypomnieć, iż samo 64 K wolnej pamięci RAM umożliwia wykorzystanie bogatego oprogramowania pracującego pod systemem **CP/M.**

W Łodzi istnieje dość liczna grupa posiadaczy tego komputera tworząca nieformalny klub jego użytkowników i posiadająca w użytkowaniu prawie 300 programów — tak gier, jak edukacyjnych i użytkowych. Wymienię chociażby różnego rodzaju Banki Danych, programy do obróbki tekstów, języki programowania **HISOFT-PASCAL, ASSEMBLER, DISASSEMBLER, FORTH,** kompilatory **BISIC-a, FORTRAN-u,** programy edukacyjne do nauki muzyki, języków obcych, matematyki i innych przedmiotów oraz mnóstwo gier przygodowych, przestrzennych, zręcznościowych i logicznych — w tym bardzo dobry program szachowy.

Polityka firmy, która utrzymuje kompatybilność produkowanych przez siebie komputerów **MZ 80A, MZ 80K, MZ 700, MZ 800** zapewnia, że każdy nowy język opracowany na którykolwiek z tych komputerów można załadować do **SHARP-a 700,** a wbudowany MONITOR identyczny w **SHARPIE**



stach (Benchmarks — Practall Computing 1985 r.) **SHARP MZ-700** plasuje się na czołowych pozycjach pod względem szybkości. Jest szybszy od **Spectrum, Commodore 64, MSX** a w niektórych testach nawet od uważanego za najszybszy BASIC na komputerach 8-bitowych **ACORN BBC,** nie wspominając o niektórych 16-bitowych jak **Texas Instruments TI-99-4A** czy **Sinclair QL.**

Oprócz firmowego **S-BASIC-a** do dyspozycji posiadaczy **SHARP-a MZ-700** jest również dostępny w Polsce **Microsoft — BASIC V.4.5** zajmujący 35 K pamięci i zapewniający obliczenia w podwójnej precyzji. Podobna wersja **BASIC-a** zaimplementowana jest w **Amstradzie CPC 6128.**

Dostępne jest także bogate oprogramowanie tworzone głównie przez firmy Software'owe w Japonii i w Europie. Głównymi producentami w Europie są firmy **„SOLO — Software"** w Wielkiej Brytanii i **„Körstner and Partner"** w RFN.

700 i **800** umożliwia rozwój własnych programów w kodzie maszynowym, a nawet dużych systemów aplikacyjnych. Dotychczas wydane zostały następujące pozycje dotyczące komputera **SHARP MZ 700:**

1. Listing cassetten BASIC 680 str.
2. Listing discetten BASIC 660 str.
3. Listing MONITOR-a (w załączonej do komputera książce) 65 str.
4. Listing Q-D BASIC 670 str.
5. Tips and Tricks for SHARP MZ-700
6. PEEKing and POKEing for SHARP MZ-700
7. Inside and Out SHARP MZ-700

Od 2 lat w Wielkiej Brytanii dla posiadaczy SHARP-ów wydawany jest specjalny kwartalnik pt. „SHARP USER NOTICE".

Jerzy Garlicki

APN SPECJALNIE DLA „BAJTKA"

Ten ośrodek naukowo-szkoleniowy rozlokował się w budynku szkoły nr 117 w Moskwie, gdzie dwa tysiące dzieci uczy się „obcować" z komputerami. Przychodzą tu uczniowie pozostałych 19 szkół dzielnicy Brieżniewskiej. Ośrodek został otwarty we wrześniu ubiegłego roku jako jeden z oddziałów Akademii Nauk ZSRR, działają tu jego pracownicy.

— Oczywiście, można by otworzyć klasy komputerów we wszystkich dwudziestu szkołach — mówi kierownik Ośrodka, kandydat nauk technicznych **Dmitrij KAZIMIROW**. — Ale jest to rozdrobnienie sił i środków. Natomiast w naszej placówce została skoncentrowana najnowocześniejsza technika i współczesna myśl naukowa. Poza tym tak jak monitor do komputera, tak i my jesteśmy podłączeni do „dużej" Akademii. Wieczorami przychodzą tu aspiranci i pracownicy naukowi, aby doskonalić swe komputerowe umiejętności. W ten sposób rozwiązujemy również inny problem — pełne wykorzystanie drogiego sprzętu ...

Ubiegłej jesieni przeszkolono tu 150 wykładowców, którzy mieli prowadzić kursy informatyki i techniki obliczeniowej w radzieckich szkołach.

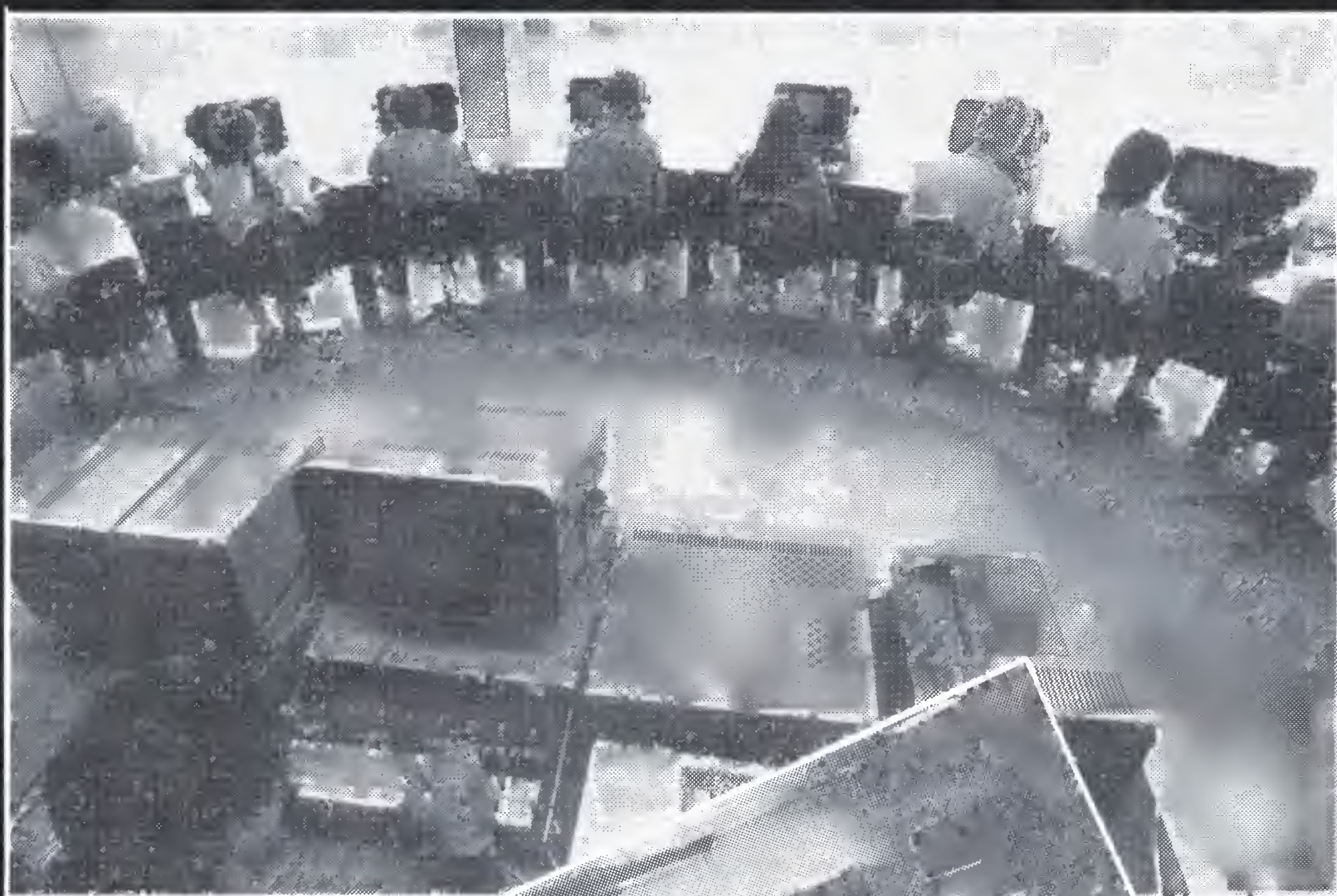
Lekcje teoretyczne w klasach 9 uczniowie odbywają w budynkach swoich szkół, a raz w miesiącu przychodzą do Ośrodka na praktyczne zajęcia z komputerami. W sprzęt komputerowy zostały wyposażone 4 sale szkolne (około stu miejsc), w dwóch z nich stoją komputery osobiste, mi.in. polskie „MERITUM".

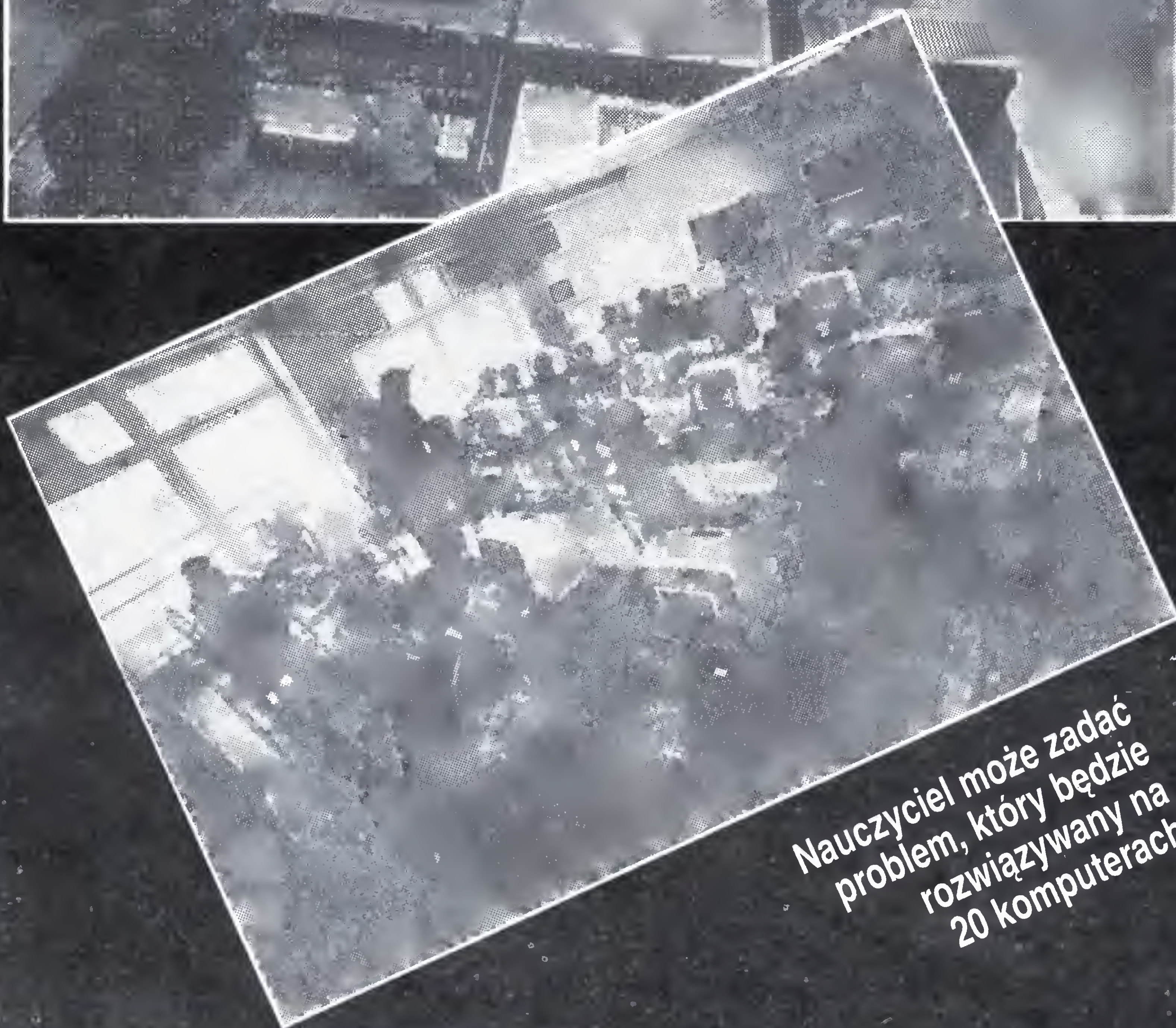
— Taka klasa dostarcza pedagogowi dużych możliwości — wyjaśnia **Dmitrij Kazimirow**. — Nauczyciel może zadać problem, który będzie rozwiązywany na 20 komputerach. Zapewnia to indywidualne podejście, uwzględniające zdolności, predyspozycje i zainteresowania uczniów. Wykładowca może również przesłać program ze swojego pulpitu do wszystkich komputerów równolegle, dać wszystkim jedno zadanie i sprawdzić wyniki.

Początkowo uczniowie rozwiązują przy pomocy komputerów najprostsze zadania szkolne, potem trudniejsze. Z każdą nową lekcją głębiej poznają naprawdę nieograniczone możliwości, jakie otwiera przed nimi „elektroniczny rozum". Często sami wymyślają sobie zadania: rysowanie wykresów lub na przykład sporządzanie tabeli mistrzostw świata w piłce nożnej. Dużą popularnością cieszą się gry komputerowe.

Szczególnie uzdolnieni mogą popracować w klasie dalekich terminali. Znajdują się tu urządzenia, połączone z ośrodkiem obliczeniowym AN ZSRR.

O ile uczniowie z innych szkół dzielnicy uczą się głównie fachowego wykorzystania komputerów, to ich koledzy ze szkoły 117 będą profesjonalnymi programistami. Już teraz, po roku zajęć, powierza się im ważne zadania — układanie programów szkolnych dla młodszych klas.





Nauczyciel może zadać problem, który będzie rozwiązywany na 20 komputerach

SZKOŁA nr 117



— Od samego początku wdrażamy młodzież do poważnej pracy — wyjaśnia Dmitrij Kazimirow. — Niczego nie robi się u nas w celach wyłącznie naukowych: ot, napisał i wyrzucił. Układane programy będą wykorzystywane do komputerowego nauczania innych przedmiotów.

Specjaliści z Ośrodka uważają, że naukę komputerowego abecadła trzeba rozpocząć nie w 9 klasie, lecz o wiele wcześniej. Po pierwsze stwierdzono, że dzieci w młodszym wieku szybciej i lepiej opanowują metody „obcowania" z komputerem. (Psychologowie są zdania, że każdy wiek posiada najbardziej sprzyjające okresy dla pewnych zajęć). Po drugie, pozwoli to realizować dalszy trudny program szkolny, opierając się na wykształceniu komputerowym.

— Mamy zamiar założyć w najbliższym czasie klub komputerowy — dzieli się planami Dmitrij Kazimirow. — Niech przychodzi, kto chce, niezależnie od wieku. Jedni zajmą się grami, drudzy — skomplikowanymi obliczeniami, trzeci — sprawdzą swoją wiedzę z różnych przedmiotów. Będzie to miało również znaczenie wychowawcze. Wiemy już, że komputer potrafi zaabsorbować dzieci i odciągnąć je od bezmyślnego spędzania czasu, bezczynności ...

Eduard Alesin

DZIŚ PISZEMY KSIĄŻKĘ

O czym? Niech będzie o piratach. Ale to nieważne. Może nawet nie będzie to książka, tylko referat, wypracowanie lub długi list. Skoncentrujemy się na technicznej stronie zagadnienia, czyli jak uwieńczyć proces twórczy czystym maszynopisem, nie zawierającym poprawek, pomyłek itd. Wbrew pozorom nie jest to problem błahy bo potencjalnych przeszkód jest bardzo wiele. Zanim wezwiemy na pomoc komputer zróbmy przegląd możliwych nieszczęść.

Po pierwsze, wszelkiego rodzaju pomyłki — uderzenie złej litery, przestawienie liter w wyrazie itp. To jeszcze nie bardzo groźne — można zamalować białą farbą i napisać poprawnie, lub zakleić kawałkiem papieru z wypisanymi właściwymi literami. Gorzej, gdy zostanie pominięty wyraz lub nawet całe zdanie. Nie ma miejsca na dopisanie i trzeba ciąć maszynopis nożyczkami i wklejać kawałki. Zrobienie tego elegancko jest bardzo pracochłonne, a często niemożliwe. Zwykle trzeba przepisywać, dobrze jeśli tylko jedną a nie kilka stron.

Ale najgorsze jeszcze przed nami. Na ogół twórca nie jest zadowolony z pierwszej wersji swojego dzieła. Po zakończeniu książki stwierdza, że w pierwszym rozdziale należy dołożyć dokładny opis statku piratów, opis brody kapitana przenieść z rozdziału czwartego do drugiego, a w całej książce słowo „kaflowy" należy bezwzględnie zmienić na „gaflowy". Więc znowu klej, nożyczki, zamalowywanie, a na koniec i tak trzeba to przepisać, i oczywiście pomylić się przy tym stosowną ilość razy. A potem jeszcze strzelbę, która strzela w ostatnim rozdziale trzeba powiesić na ścianie na początku pierwszego. Więc znowu klej, nożyczki ...

No dobrze, sytuacja jest kiepska, ale gdzie tu komputer? Otóż pomysł jest bardzo prosty, jak zresztą większość pomysłów na gnębienie pracą tego bezbronnego stworzonka. Zapisujemy tekst nie na papierze, tylko w pamięci operacyjnej. Tam jest dużo łatwiej zamazywać, poprawiać, przepisywać. A gdy tekst będzie już piękny i gotowy to przelejemy go na papier za pomocą drukarki.

Żeby zaspokoić nasze rozliczne wymagania potrzebny będzie jeszcze program. Po angielsku taki program najczęściej nazywa się „wordproccessor", po polsku — „edytor tekstów" lub też „program redakcyjny".

Jakie możliwości dają użytkownikowi programy redakcyjne dostępne na współczesnych mikrokomputerach? Po pierwsze ogromne, po drugie już je w pewnym sensie wyliczyliśmy, mówiąc o trudnościach czyhających na autora „Przygód Jednogłowego Pirata". Ale nazwijmy te możliwości po imieniu.

Podstawa pracy, to wczytanie wprowadzanego z klawiatury tekstu i wyświetlenie go na ekranie. Dalej, zapisanie tekstu w pamięci zewnętrznej i możliwość ponownego wczytania do dalszej obróbki. Możliwość łatwego ustawienia kursora w dowolnym miejscu wyświetlonego na ekranie tekstu i wprowadzenia tam poprawek — zmiany wpisanych znaków, ich usuwania i dostawiania. Oczywiście przy kasowaniu znaków pozostała część tekstu przesuwa się w lewo — na wolne miejsce, a przy wstawianiu w prawo — aby zrobić miejsce na nowe znaki. Jak to jest zrobione — to nie obchodzi pisarza, o to martwił się programista.

Ale to dopiero początek. Następna poważna grupa możliwości to operowanie fragmentami tekstu. Dowolny kawałek można zaznaczyć (wskazując kursorem na jego początek i koniec) i przenieść w całości w inne miejsce. Albo usunąć, lub wypisać na dyskietkę jego odrębną całość. Można też dopisać do naszego tekstu inny, uprzednio przygotowany i zapisany na dyskietce dokument.

Na ogół wszystkie funkcje wywołuje się bardzo prosto — jednym lub kilkoma uderzeniami wybranych klawiszy funkcyjnych. Jeśli ktoś nie pamięta co znaczą które klawisze, to może zażądać stałego wyświetlania legendy gdzieś w kącie ekranu.

Tekst zwykle wielokrotnie przekracza pojemność ekranu, więc mamy grupę komend pozwalających łatwo odszukać i wyświetlić potrzebny fragment. Typowe to: na początek/koniec tekstu, przesuń o stronę w przód/tył, znajdź w tekście zadany ciąg znaków i ustaw na nim kursor.

Dalej mamy możliwość zastąpienia wskazanego ciągu znaków innym ciągiem, w jednym miejscu lub wszędzie, w całym dokumencie. Np. jedną komendą można spowodować zamianę w całym tekście słowa "woda" na słowo "ściek". (Uwaga, żadne udogodnienia nie zwalniają od myślenia! Komputer nie rozumie tekstu i jeśli gdzieś było napisane "woda ognista", to tam też zamieni i będzie "ściek ognista". Oczywiście i na to znaleziono radę — można rozkazać: "Przed każdą zamianą wyświetl wynik i zapytaj o zgodę". Ale i tak uważać trzeba zawsze).

No i wreszcie drukowanie. Pisząc nie musimy się martwić o wygląd gotowego tekstu. Załatwimy to komendami ustalającymi format wydruku. Raz wprowadzony do pamięci tekst możemy drukować w najrozmaitszej postaci. Komputer wyrówna (na życzenie,) prawy margines, wydrukuje zadane podkreślenia i wytłuszczenia, podzieli na strony. W razie potrzeby wydrukuje podane nagłówki i ponumeruje strony. A jeśli stwierdzimy, że np. odstępy między liniami są za małe, lub margines za szeroki, to wystarczy zmienić kilka komend i po prostu wydrukować od nowa.

Na mikrokomputerach trochę lepszych od Spectrum można spotkać edytory współpracujące z programami znajdującymi błędy ortograficzne. Taki program, wyposażony w słownik zawierający kilkadziesiąt lub nawet kilkaset tysięcy słów, potrafi wykryć znaczną większość popełnionych przy pisaniu błędów. Podkreślam — tylko błędów ortograficznych, czyli niepoprawnie napisanych słów. Niestety, większość z tych programów robi to tylko dla tekstów w języku angielskim.

Skoro już padło słowo „język", to jesteśmy blisko bardzo dla nas poważnego problemu, to znaczy liter specyficznych dla języka polskiego (ą, ę itd.). Programy, które są w tej chwili najbardziej rozpowszechnione w Polsce, pochodzą z innych stref językowych i nie znają polskich znaków. Ale sytuacja powoli się poprawia. Część działających u nas firm informatycznych zaczyna oferować oprogramowanie, a nawet przeróbki sprzętowe, pozwalające pisać polskimi literami.

I jeszcze na koniec krótka dyskusja ze sceptykami. Czy rzeczywiście programy redagujące teksty są komuś potrzebne? Chyba tak, ja sam znam sporo osób, które ich używają. A Wy znacie przynajmniej jedną, bo oryginał tego artykułu też powstał przy pomocy mikrokomputera.

Andrzej Pilaszek



„HORACY" W KUTNIE

Dość liczna grupa ludzi w Kutnie ma komputery osobiste. Początkowo myśleliśmy o prywatnym kole zainteresowań. Ówczesny przewodniczący zarządu miejskiego ZSMP Mirek Kucharski wysunął propozycję, żeby utworzyć klub młodzieżowy. „Horacy" powstał 16 września 1985 roku. Do stycznia nie mieliśmy żadnego sprzętu.

Pomieszczenie o powierzchni 60 m kw. mamy wypożyczone w liceum ogólnokształcącym im. J. Kasprowicza. Ponieważ odbywają się w nim lekcje, sprzęt musi być codziennie chowany. Z tego względu gdyby była okazja poszukalibyśmy sobie innego lokum.

— Zwróciliśmy się do różnych zakładów z propozycjami pomocy. W statucie mamy klauzulę, że sprzęt wypożyczony lub przekazany przez zakład pozostaje jego własnością. Gdyby nasz klub rozpadł się, sprzęt powróci do ofiarodawców. Największą pomoc okazały nam kutnowskie zakłady farmaceutyczne „Polfa". Otrzymaliśmy z „Polfy" trzy Spectrum plus i Commodore ze stacją dysków, a nasza odlewnia zakupiła w firmie polonijnej Unipolbrit 2086. Mamy również magnetofony z Miflexu, a zawodowa straż pożarna przekazała nam kolorowy telewizor.

Ze względu na małą ilość sprzętu chwilowo wstrzymaliśmy nabór członków, wg stanu z kwietnia mamy ich 104.

Na razie prowadziliśmy zajęcia szkoleniowe w 5 grupach. Każda grupa spotykała się raz w tygodniu. Mieliśmy obiecany etat z TMMT, niestety, nie wypalił. Przy obsadzie etatowej klub mógłby być dłużej otwarty.

Na cztery godziny zajęć — 1,5 godziny przeznaczamy na gry. Podczas szkolenia prowadziliśmy język BASIC na Spectrum.

— Zachodniej literatury nie posiadamy. Opieramy się głównie na „Informatyce", „Bajtku" i „Komputerze". Oprócz tych czasopism mamy też kilka egzemplarzy opisu języka Logo.

— Mamy ok. 250 programów, 80 proc. stanowią gry. Resztę — programy matematyczne, nauki j. angielskiego, z użytkowych — edytory tekstu, język PASCAL i FORTH.

— Z TMMT obiecano nam dwa Meritum, lecz dostawa z Zabrza opóźnia się.

Naszym najbliższym celem jest zdobycie drukarki, dzięki której moglibyśmy na przykład prowadzić kartotekę członków klubu, a tak klub jest prowadzony na papierach. Chcemy również rozszerzyć konfigurację sprzętu o stacje dyskietek. Marzy nam się stworzenie sieci komputerowej w samym klubie.

Stefan Jagiełło
przewodniczący Wojewódzkiego Klubu Komputerowego ZSMP „Horacy" w Kutnie

„INFORMIK" W WARSZAWIE

Warszawski Klub Mikroinformatyczny **„InforMik"** powstał 1.IX.85 r. Klub **„InforMik"** jest klubem otwartym do którego może należeć każdy w wieku od lat 14-tu. Nasz klub stawia sobie takie cele jak:

- Popularyzacja mikroinformatyki i elektroniki w szerokich kręgach młodzieży szkolnej, zwłaszcza zaś w środowisku szkół średnich.
- Zapewnienie uzdolnionym i zaangażowanym uczniom dogodnych warunków dla szybkiego rozwoju talentów i wzbogacenia wiedzy. Jak i do intensywnej pracy nad sobą oraz rozbudowanie aktywności społecznej.
- Stworzenie warunków do rozwijania wszechstronnej twórczości programistycznej wśród młodzieży w szczególnym uwzględnieniu zastosowań techniki mikrokomputerowej w edukacji i samokształceniu.
- Integracja środowiska entuzjastów nowoczesnej elektroniki i informatyki wokół problemu rozwoju zainteresowań i uzdolnień technicznych. I właśnie na takich celach pragniemy opierać swoją działalność.

Warszawski **„InforMik"** brał udział w wielu pokazach komputerowych na terenie całej Polski. Z komputerami (własnymi), byliśmy w:

a) Warszawa — Mikroexpo'85, Infovideo'86
b) Wałbrzych — Komputery'86

Naszym sponsorem jest WSM — Wawrzyszew.

W naszym klubie oprócz członków rzeczywistych i członków kandydatów są ludzie, którzy bardzo nam pomagają przetrwać ten ciężki okres, i właśnie takimi ludźmi są członkowie honorowi naszego klubu. Wśród nich są: **Andrzej Kurek, Tomasz Pyć, Waldemar Olszewski, Piotr Tymochowicz** i wielu innych.

Nasz klub współpracuje z takimi klubami jak: Abakus, Maniak, Mikros oraz klub użytkowników Atari w Krakowie. Posiadamy dużą bibliotekę programów na takie komputery jak: C-64, C-128, Spectrum, Meritum, Atari, Oric, Laser i wiele innych.

Oprócz biblioteki programów posiadamy dużo literatury fachowej takiej jak: Instrukcja do C-64 po polsku, poradnik młodego Hackera itp. Mamy nadzieję, że nasza biblioteka będzie się powiększać. Chętnych, którzy chcieliby się skontaktować z nami prosimy o wysyłanie listów na adres:
Jarosław Jaworski ul. Andersena 5 m 49 01-911 Warszawa.

Prezes W-wskiego Klubu InforMik
Jarosław Jaworski

OGŁASZAMY WSPÓŁZAWODNICTWO KLUBÓW MIKROKOMPUTEROWYCH

O ZŁOTĄ DYSKIETKĘ BAJTKA

Jak powinien być zorganizowany dobry klub mikrokomputerowy? Powinien być dostępny dla wszystkich, a może tylko dla tych, którzy wykażą się wpierw działalnością i zasłużą sobie na to wyróżnienie? Jak zapobiec zamienianiu się klubów w salony gier komputerowych? Czy składka członkowska jest potrzebna, czy klub może zarobić na siebie sam?

Entuzjazm jest warunkiem koniecznym każdej skutecznej działalności. Nie jest jednak — niestety — warunkiem wystarczającym. Najwyższy więc czas zastanowić się wspólnie jak powinien wyglądać klub mikrokomputerowy aby jego działalność była możliwie najbardziej efektywna.

Najbardziej kompetentni w tej sprawie są oczywiście sami organizatorzy klubów. Zapraszamy Was do dyskusji.

Jednocześnie — zgodnie z zapowiedzią — ogłaszamy współzawodnictwo Klubów o Złotą Dyskietkę „Bajtka".

Już wkrótce opublikujemy w „Bajtku" kupon zgłoszenia do konkursu. Mamy nadzieję, że przykłady najlepiej zorganizowanych kubów będą wzorem dla tych, które dopiero rozpoczynają działalność.

Bajtek

LISTY DO MARKA L.

W 5-6 numerze Bajtka opublikowaliśmy fragment listu Marka L. — człowieka niepełnosprawnego proszącego o pomoc w uzyskaniu oprogramowania do swojego ZX Spectrum. W kilka dni po ukazaniu się numeru w sprzedaży otrzymaliśmy w tej sprawie bardzo wiele listów. Niezmiernie cieszy fakt, że ów krótki apel wystarczył do utworzenia łańcucha ludzi dobrej woli. Oto fragmenty niektórych odpowiedzi:

W odpowiedzi na apel Pana Marka L. zgłaszam swoją gotowość do pomocy w zdobyciu programów do Spectrum. Utrzymuję kontakty niemal z całą Polską z osobami, które chcą współpracować i wymieniać się oprogramowaniem, literaturą, mapami gier oraz instrukcjami. Proszę zatem o przekazanie mojego listu Panu Markowi. Jeżeli chce uzyskać ciekawe programy, to niech napisze i prześle na podany adres kasetę magnetofonową. Proszę, żeby podał, jakie gry go interesują lub jakie chciałby otrzymać. Gwarantuję, że w ciągu tygodnia Pan Marek otrzyma nagraną taśmę z programami (tydzień liczę od chwili otrzymania taśmy). Oto mój adres:

Andrzej Piechowski
Os. Glinnik bl. 11 m 7
97-205 Tomaszów Maz. 7

Proszę o podanie mi adresu p. Marka L. (nr 5-6 „Bajtka") lub przesłanie mu tej kartki. Chciałbym zaofiarować mu — nieodpłatnie, rzecz jasna — nieco dobrego oprogramowania.

Sławomir Maculewicz
ul. Wyspiańskiego 5 DS-9
80-434 Gdańsk-Wrzeszcz

Nazywam się Dariusz Sarna, mam 16 lat i jestem uczniem Technikum Mechanicznego w Elblągu. Mam szeroki dostęp do gier i programów użytkowych na ZX Spectrum i chętnie pomogę Panu Markowi L., którego list wydrukowaliście w „Bajtku".

Dariusz Sarna
ul. płk. Dąbka 69 m 51
82-300 Elbląg

... i tak dalej — nadeszło jeszcze wiele listów o podobnej treści.

Prosimy teraz Pana Marka o kontakt listowny z naszą redakcją. Przekażemy mu całość nadesłanej w tej sprawie korespondencji.

Jednocześnie tak szeroki oddźwięk nasunął nam następujący pomysł: prawdopodobnie osób niepełnosprawnych, posiadających własny mikrokomputer i kłopoty w zdobywaniu doświadczeń jest w naszym kraju więcej. Czy możliwe byłoby — w oparciu o dobrą wolę naszych czytelników — stworzenie banku pomocy takim osobom? Sądzimy, że tak! Wszystkich zainteresowanych, zarówno potrzebujących pomocy, jak mających możliwości i pragnących jej udzielić, prosimy o listy.



Drogi Bajtku!

Na listy czytelników odpowiada Marcin Waligórski Student III roku informatyki UW

Nie wiem, czyja to wina, czy redagujących czy wydających, ale prawie każdy z podanych w waszym czasopiśmie programów jest błędny. Jestem posiadaczem ZX Spectrum Plus i spędzam przy nim wiele czasu. Ponieważ zajmuję się nie tylko grami, często wprowadzam i zapoznaję się z różnymi programami, co w przypadku wydawnictw krajowych jest kompletną stratą czasu, pieniędzy i zbędnym użytkowaniem sprzętu. Na poparcie moich zarzutów załączam wykaz programów, których nie można wpisać ze względu na zawarte w nich błędy:

— Bajtek nr 2, str. 30 W POSZUKIWANIU BLIŹNIAKA ZIEMI
str. 29 LICZYDEŁKO

— Bajtek nr 3, str. 31 — ROZMÓWKA

Mam nadzieję, że redakcja nie potraktuje mojego listu jako złośliwość, lecz jako głos czytelnika i wpłynie na poprawę jakości swoich wydawnictw.

Pozostaję z szacunkiem

Cezary Sądek
ul. Derdowskiego 22/21
71-087 Szczecin

Program „W poszukiwaniu bliźniaka ziemi" został napisany i uruchomiony na Spectrum. Wydruk zamieszczony numerze 2 jest kopią działającego programu, zatem mamy podstawę by twierdzić, że jest on prawidłowy.

Aby uruchomić na Spectrum programy z artykułów „Liczydełko", trzeba trochę znać Sinclair Basic. W „Liczydełku" trzeba skrócić nazwy zmiennych łańcuchowych do jednej litery (plus znak $). Zamiast ZNAK$
napiszemy zatem np.
Z$

W programie „Rozmówka" oprócz powyższego trzeba zamienić nie istniejącą w tym dialekcie Basica funkcję RIGHT$ na
I$((LEN (I$) — 1) TO LEN (I$))
— zmiana nastąpić powinna w linii 30.
Poza tym należy wziąć pod uwagę, że Spectrum w nieco inny sposób formatuje wydruki na ekranie niż Amstrad, na którym program ten był uruchomiony. Otóż np. w przypadku użycia.

PRINT „Wyglądasz na "; wiek.
pomiędzy słowo „na" a wypisywaną wartość liczby wiek nie zostanie wstawiona spacja. Należy zatem napisać.
PRINT „Wyglądasz na" ; wiek
Sądzę, że wprowadzenie programu z tymi poprawkami powiedzie się.

Lublin 12.07.1986 r.

Redakcja "BAJTEK"
ul. Wspólna 61
00-950 Warszawa

Szanowna Redakcjo,

Serdecznie dziękuję za bardzo życzliwe potraktowanie mojej prośby i przesłanie cennych sugestii pozwalających na wybór mikrokomputera umożliwiającego realizację moich zamierzeń naukowo-badawczych. Bardzo szczegółowe odpowiedzi na pytania, zawarte w Twoim liście, w pełni mnie satysfakcjonują.

Łączę wyrazy szacunku

Witold Ko[illegible]
lekarz medycyny

Obecnie redakcja nasza przestrzega dość rygorystycznie zasady publikowania programów w postaci kopii wydruków z drukarki (o czym już kiedyś wspominałem) co daje nam w większości wypadków gwarancję poprawności publikowanego tekstu programu.

Okazuje się też, że w praktyce przy tym sposobie „transmisji" programu, o którym mówimy, najbardziej zawodnym ogniwem okazuje się... człowiek, wpisujący program na komputer. Pomimo pewnej praktyki, nigdy nie udało się mi przepisać od razu bezbłędnie programu liczącego więcej od 30 do 40 linii, niezależnie od tego, w jakim języku został on napisany.

Nikt nie jest jednak nieomylny i w przypadku faktycznego stwierdzenia błędu w opublikowanym przez nas programie prosimy o zawiadomienie nas o tym, podając wszakże, o jakiego rodzaju błąd chodzi.

Czy została wydana gra DRAGONTORC na COMMODORE 64?

Bartosz Makowski
ul. Zagłoby 35 m 13
02-495 Warszawa

Tak. Gra ta dostępna jest również na Amstradzie, Spectrum 48 kB oraz — w wersji dyskowej — na Atari 800 XL

Chciałbym się trochę więcej dowiedzieć o komputerze domowym AMIGA, o którym przeczytałem w 3/4 n-rze „Bajtka". Jaka jest sytuacja z oprogramowaniem na ten komputer?

Roman Zadroga
Plac Wolności 16/3
18-500 Kolno

Oto kilka danych technicznych maszyny Commodore-Amiga o której tak głośno było na zachodzie Europy w pierwszych miesiącach bieżącego roku.

Jednostka centralna: Motorola 68000 — zegar o częstotliwości 7,16 MHz

Pamięć RAM 512 kB

Pamięć zewnętrzna: wbudowana stacja dysków o średnicy 3 1/2 cala z możliwością pomieszczenia 880 kB danych na jednym dysku. Możliwość podłączenia następnych stacji dysków (również 5 1/4 — calowych) oraz dysku twardego 20 MB.

Gniazda połączeniowe: możliwość dołączenia dodatkowych modułów pamięci. Zabudowane interface 'y standartu Centronics i RS 232. Dwa gniazda manipulatorów. Możliwość podłączenia myszy, modemu, wzmacniacza stereofonicznego.

Ekran tekstowy: 40, 60 lub 80 znaków w wierszu

Grafika: Rozdzielczość ekranu 640 × 400 punktów z możliwością jednoczesnego użycia 16 spośród 4096 kolorów lub 320 × 400 punktów z paletą 32 kolorów.

Dźwięk: programowy syntezator stereofoniczny możliwość generowania mowy.

W chwili obecnej na rynku zachodnim dostępna jest już nie budząca większych zastrzeżeń ilość programów dla komputera Amiga. To niektóre z nich: wśród programów użytkowych —
Deluxe Paint, Deluxe Music Construction Set,
Deluxe Printing firmy Electronic Arts,
Aegis Images i Aegis Animat — f-my Island Graphics.
Wśród programów narzędziowych — języki programowania: C (firmy Lattice), Turbo Pascal (Borland), Cambrigde LISP (Metacomco) oraz ABasic (interpreter i kompilator firmy Metacomco). Dostępna jest również baza danych DBase III oraz emulator IBM firmy Simile Resarch.

Do listy tej dodać można jeszcze kilkadziesiąt programów przydatnych w pracy biurowej oraz gier.

Interesuje mnie możliwość podłączenia modemu do mojego C — 64. Jak kształtuje się obecnie cena takiego urządzenia na zachodzie? Chciałbym poznać również typy modemów obecnie spotykanych na rynku.

J.M.
(nazwisko i adres do wiad. redakcji)

Rozpiętość cen produkowanych obecnie modemów jest spora i sięga granic 200 — 1000 DM. Koszt urządzenia średniej klasy plasuje się mniej więcej w środku podanego zakresu, tzn. 300-600 DM. Niektóre typy modemów mogących współpracować z Commodore 64 to: Epson CX-21, Hitrans 300, Dataphon S 21 D, CTIK 2002, Ascom AS-8001 (najtańszy z wymienionej grupy). Powyższe informacje podaję za miesięcznikiem „64'er" z marca 1986 r.

*Kolega sprezentował mi niedawno rozbudowany **Basic** na swoim Amstradzie. Sam posiadam „tylko" Spectrum. Czy można przy pomocy Basica tego komputera uzyskać odpowiedniki instrukcji WHILE, WEND, ELSE — które ułatwiają pisanie programu.*

Piotr Osiecki
(adres do wiad. redakcji)

Pętlę WHILE (dopóki...) da się dość łatwo, aczkolwiek mało czytelnie, zastąpić używając instrukcji skoku. Przypuśćmy mianowicie, że mamy następującą pętlę

```
1010 WHILE warunek
1020 REM Tutaj instrukcje wykonywane w petli
....
1990 WEND
2000 REM Tutaj dalsza czesc programu
....
```

Oto nasz fragment programu przeniesiony na Spectrum

```
1010 IF NOT warunek THEN GOTO 2000
1020 REM Tutaj instrukcje wykonywane w petli
....
1990 GOTO 1010
2000 REM Tutaj dalsza czesc programu
....
```

Słowo ELSE jest częścią rozbudowanej instrukcji warunkowej IF... THEN... ELSE... (Jeżeli... to..., w przeciwnym przypadku...) W uboższych odmianach Basica można zastąpić taką konstrukcję również przy pomocy skoków i znów ze stratą czytelności programu. Zamiana może mieć np. postać taką.

```
1010 IF NOT warunek THEN GOTO 1100
1020 REM Tutaj instrukcje wykonywane,
1030 REM gdy warunek jest spelniony
....
1090 GOTO 1200
1100 REM Tutaj instrukcje wykonywane,
1110 REM gdy warunek jest niespelniony
....
1200 REM Tutaj dalsza czesc programu
```

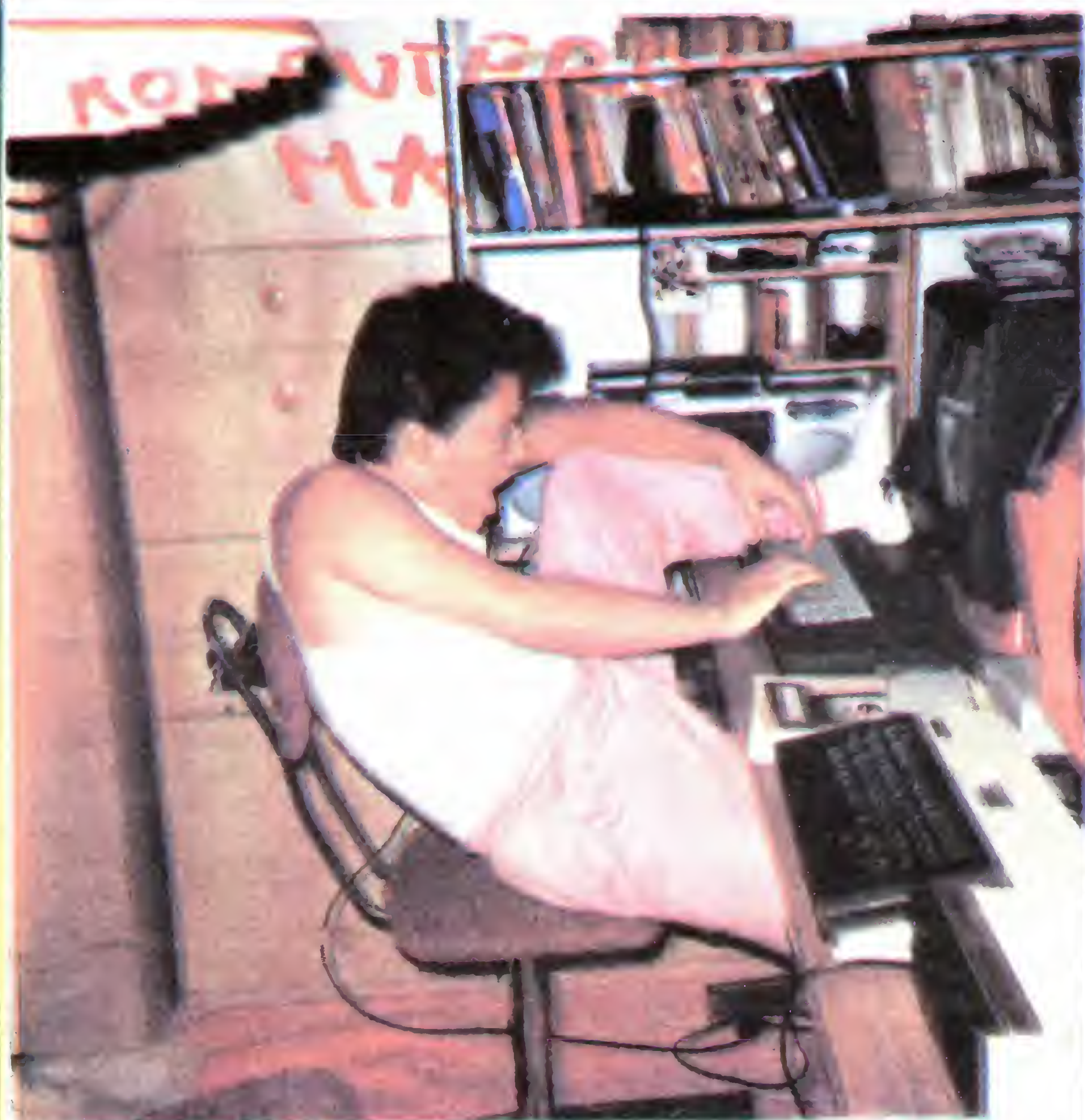
Marcin

Prosimy o opatrywanie listów dopiskiem na kopercie w zależności od tego, z jakim działem naszego czasopisma związana jest ich tematyka. Proponujemy następujące dopiski: COMMODORE, SPECTRUM, ATARI, MERITUM, AMSTRAD, KLAN NIETYPOWYCH — na listach kierowanych do poszczególnych „klanów", LISTA PRZEBOJÓW — tak jak dotychczas, na kartkach i listach z głosami do listy przebojów gier.

SPRZĘŻENIE ZWROTNE — na listach adresowanych do niniejszej rubryki.

Oczywiście listy o innej tematyce, czy też bez owych dopisków, będą w dalszym ciągu z uwagą czytane i brane pod uwagę przy redagowaniu „Bajtka".

INDYWIDUALNY BANK DANYCH



TO JA! TOMEK...

Posiadam mikrokomputery: Atari 800 XL i Schneider CPC 6128 oraz stację dyskietek Atari, magnetofon Atari, monitor kolorowy, zestaw drążków sterowych i piór świetlnych.

Interesuję się mikrokomputeryzacją, muzyką elektroniczną, fantastyką oraz wszystkim, co związane jest z komputerami, grafiką i plastyką.

Posiadam także następujące programy własne:
- **oprogramowanie edukacyjne**
- **programy muzyczne, graficzne i demonstracyjne**
- **programy narzędziowe**

Swoje programy sygnuję napisem © VON GEN'S.

Innym użytkownikom mogę zaproponować oprogramowanie na komputery Atari i Schneider, literaturę i czasopisma oraz wiele innych ciekawostek komputerowych.

Jestem współzałożycielem i prezesem ATARI KLUB przy Zespole Szkół Hutniczo-Mechanicznych w Sosnowcu. Posiadamy pięć komputerów ATARI 800 XL i jeden Schneider CPC 6128. Prócz własnych programów dysponujemy pełnym zestawem programów dostępnych w Polsce.

Tomasz Piątkowski (lat 17)
41-200 Sosnowiec
ul. Mielczarskiego 24/18

Posiadam mikrokomputery ZX Spectrum i Spectrum+ oraz magnetofon i drukarkę. Interesuję się wszystkim co dotyczy komputerów. Posiadam także własne programy użytkowe i parę gier

Ryszard Kajstura (lat 12)
ul. Szymanowskiego 1/21
43-400 Cieszyn

Posiadam mikrokomputer TIMEX 2048 (kompatybilny z ZX Spectrum) oraz magnetofon i telewizor. Interesuję się komputerami, muzyką i sportem.

Proponuję wymianę doświadczeń, literatury, programów. Chciałbym poznać programy użytkowe, z którymi nie miałem kontaktu.

Andrzej Jędrychowski (lat 14)
II MPS 16/105
Gdynia-Witomino

Posiadam mikrokomputer SHARP 731 oraz drukarkę i magnetofon wmontowane w komputer. Interesuję się science-fiction, komputerami, filmami i literaturą. Wymienię się grami i programami dydaktycznymi ewentualnie zamienię komputer SHARP na Spectrum +

Lesław Król (elektryk, lat 50)
ul. Solskiego 3/37
44-200 Gliwice

Posiadam mikrokomputer ZX Spectrum 48K. Interesuję się komputerami, literaturą, hafciarstwem, przyrodą i psami. Chciałabym poznać dzieci początkujące tak jak ja i bardziej zaawansowane, oraz nowe możliwości komputera.

Magdalena Bartkowiak (lat 10)
ul. Krakusa 14/32
41-500 Chorzów

Posiadam mikrokomputer MICRO-PROFFESOR II (MPF II) oraz drukarkę termiczną MPF II i dodatkową klawiaturę. Interesuję się komputerami i programowaniem. Posiadam także następujące własne programy:
- **średnia ważona**
- **rachunek błędów (średnie odchylenie standartowe)**
- **liczby pierwsze**
- **inne specjalistyczne**

Wymienię się programami na kasetach lub wydrukami.

Michał Głębowski (lat 12)
ul. Lniana 6 m 74
91-158 Łódź

Posiadam mikrokomputer ORIC-Atmos 48k oraz monitor telewizor PAL-SECAM, magnetofon dwukasetowy. Interesuję się elektroniką i informatyką. Posiadam także następujące własne programy:
- **gry zręcznościowe**
- **nauka matematyki (działania podstawowe dla dzieci)**
- **wyszukiwanie opisów, schematów urządzeń elektronicznych (telewizorów, magnetofonów różnego typu itd.)**
- **rysowanie przy pomocy komputera**

Innym użytkownikom proponuję współpracę w programowaniu i opracowywaniu nowych programów komputerowych.

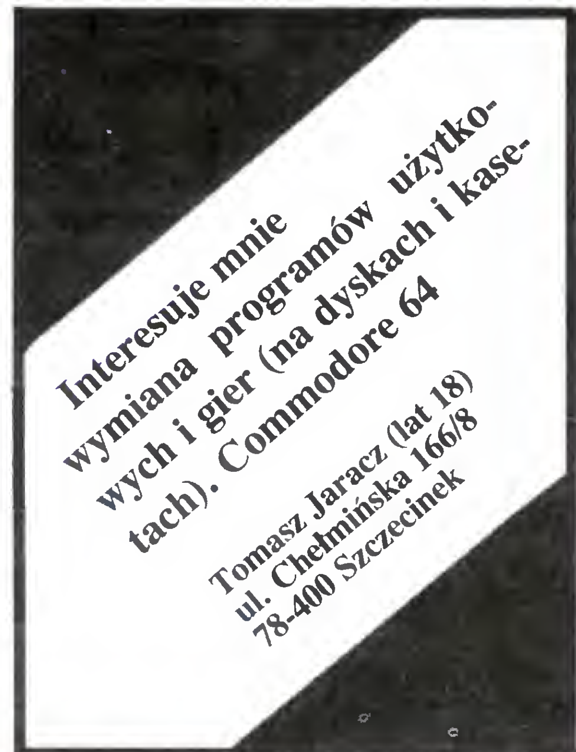
Janusz Helbin
(elektronik, lat 32)
ul. Łużycka 53 m 209
30-658 Kraków

Posiadam mikrokomputer Commodore 64 oraz Datassette 1541. Interesuję się informatyką, elektroniką, muzyką, mechaniką samochodową i majsterkowaniem. Chętnie wymienię się na programy edukacyjne.

Piotr Mięsowicz
(teletechnik, lat 27)
ul. Bolesława Śmiałego 49
43-190 Mikołów

Posiadam mikrokomputer Atari 800XL oraz magnetofon Atari XC 12 i drążek sterowy. Interesuję się informatyką, elektroniką i sportem. Udostępnię lub wymienię programy i gry.

Robert Jeziorski (lat 16)
ul. Mostnika 13 m 2
76-200 Słupsk



Posiadam mikrokomputer Atari 800XL oraz magnetofon Atari-1010.

Interesuję się krótkofalarstwem (znak SP7FUZ). Posiadam program RTTY (radiodalekopis)

Henryk Gajecki
(pedagog, lat 40)
Os. Waryńskiego bl. 6 m 5
98-300 Wieluń

Posiadam mikrokomputer

VC-20

i 300 programów.
Szukam kontaktu z posiadaczami VC-20.

Wojtek Sławiński
ul. Burgaska 5 m 11
02-758 Warszawa

Posiadam mikrokomputer Atari 800XL oraz magnetofon Atari-1010. Interesuję się informatyką i matematyką. Posiadam także następujące własne programy:

- tabliczka mnożenia
- ortografia (nieco uboższa od programu J. Potempy)
- figury
- praca domowa
- piorun (ulepszona wersja firmowa)
- rozwiązywanie równań
- rozwiązywanie układów równań
- program na znajdowanie NWD liczb i inne

Wymienię się oprogramowaniem i informacjami

Daniel Cichocki (lat 13)
ul. Bohaterów Studzianek 4/15
26-900 Kozienice

UWAGA!

Ośrodek Postępu Technicznego w Katowicach pod patronatem Urzędu Postępu Naukowo-Technicznego i Wdrożeń, Zrzeszenia MERA i PTI organizuje w dniach od 1986-11-17 do 1986-11-21 na swoim terenie Ogólnopolskie Targi Oprogramowania SOFTARG' 86 (OTO SOFTARG' 86). Na Targach prezentowane będą systemy i programy przeznaczone dla następujących rodzin komputerów:

— Komputer Jednolitego Systemu (Riad, IBM 360/370),
— Komputer typu SM (SM-4, Mera-60, PDP 11, SM 1300),
— IBM PC (ComPAN-16, Elwro 800, Olivetti M-24, in.),
— mikrokomputery personalne i biurowe (Meritum, Spectrum, Schneider (Amstrad i in.).

Wszystkie produkty programowe wystawiane na OTO SOFTARG' 86 umieszczone będą w katalogu dostępnym w czasie trwania i po zakończeniu Targów (koszt 1 egz. — 5.000.- zł.).
Zapraszamy do wzięcia udziału w OTO SOFTARG' 86.

INFORMACJI SZCZEGÓŁOWYCH UDZIELA:



Ośrodek Postępu Technicznego
ul. M. Buczka 1b
40-955 Katowice 2
tel. 59-60-61 wewn. 264.

K-144

NOWO OTWARTY

Skład Konsygnacyjny „COMPEX"
Marszałkowska 60 przy PHZ „DYNAMO"

■ **oferuje do sprzedaży za waluty wymienialne szeroki asortyment części i podzespołów elektronicznych**

DO

- komputerów
- drukarek
- sprzętu Hi-Fi
- VIDEO-REKORDERÓW
- telewizorów itp.
- przyrządy pomiarowe
- narzędzia serwisowe
- Możliwość sprzedaży katalogowej

tel. 28-38-23.

D-112

amico

AGENCJA KOMPUTEROWA
SOSNOWIEC

PROGRAMY KOMPUTEROWE POCZTĄ, dla ATARI, AMSTRADA, COMMODORA, i SPECTRUM wysyła Agencja Mikrokomputerowa, Sosnowiec P-157, tel. 699-649.

K-76

JAK REKLAMOWAĆ SIĘ W BAJTKU?

Reklamy przyjmuje Młodzieżowa Agencja Wydawnicza (Redakcja Wydawnictw Poradniczych i Reklamy), 04-028 Warszawa, Al. Stanów Zjednoczonych 53, pokój 313. tel. 10-56-82

Cena ogłoszeń: 200 zł za 1 cm plus dodatki za kolor

przedsiębiorstwo polonijno-zagraniczne 
02 658 warszawa, ul. filona 16, tel. 43 03 84, 43 75 66, 43 93 41, tlx 817218

oferuje:

√ profesjonalny mikrokomputer **imp 85 m**
mikroprocesor Intel 8085, RAM 64 kB, dyski elastyczne 8 cali 2 × 640 kB, interfejsy DZM i V-24, system operacyjny CP/M 2.2 oraz kompilatory, bazy danych, edytory itp.
* dostawa w IV kw. 1986 r.

√ profesjonalny mikrokomputer **imp 85 m+**
mikroprocesor Intel 8085 z pamięcią RAM 256 lub 512 kB, dyski elastyczne 8 cali 2 × 640 kB, dysk elektroniczny, grafika, system operacyjny CP/M+ i bogate oprogramowanie.
* dostawa od I kw. 1987 r.

√ mikrokomputer **imp 85 w+**
czterostanowiskowa odmiana **imp 85 m+**: zegar 8 MHz, z systemem operacyjnym wielodostępnym MP/M II. Na życzenie dysk sztywny 20 MB.
* dostawa od I kw. 1987 r.

√ mikrokomputer osobisty **imz 80**
mikroprocesor Z 80A, RAM 64 kB, minidyski elastyczne, zestaw interfejsów, system operacyjny CP/M 2.2, kompilatory itp.
* dostawa w IV kw. 1986 r.

√ mikrokomputer osobisty **imz 80 m**
rozszerzona wersja **imz 80** z pamięcią RAM 256 kB i dyskiem elektronicznym.
* dostawa od I kw. 1987 r.

√ profesjonalny mikrokomputer 16-bitowy **imp 86**
odpowiednik IBM PC z pamięcią RAM 640 kB, monitorem graficznym monochromatycznym 640 × 200 punktów, dyskiem sztywnym 20 MB i bogatym oprogramowaniem.
* dostawa w IV kw. 1986 r.

√ system **master**
karta i oprogramowanie umożliwiające realizację wielodostępu na mikrokomputerach klasy IBM PC/XT.
* dostawa od I kw. 1987 r.

√ zdalny monitor ekranowy **imp 8502 m**
terminal do EMC serii Odra 1300 i ICL 1900, emulator monitora ICL 7181/2.
* dostawa od I kw. 1987 r.

√ **tsla**
przezroczysty adapter interfejsu V-24 umożliwiający pracę 10 monitorów **imp 8502 m** równocześnie przez jeden modem.
* dostawa od IV kw. 1986 r.

√ **qsla**
kolejkujący adapter interfejsu V-24 umożliwiający pracę 8 monitorów **imp 8502 m** przez jeden modem, realizuje algorytmy, pozwalające bardziej efektownie wykorzystać linię i czas jednostki centralnej.
* dostawa od IV kw. 1986 r.

√ **pc 8/16**
16-bitówka do mikrokomputerów 8-bitowych, posiada mikroprocesor Intel 8088, RAM 768 kB, i oprogramowanie emulujące IBM PC.
* dostawa wersji do: **imp 85 imz 80** - od IV kw. 1986 r.
ELWRO 500, MERA 60 - od I kw. 1987 r.
MERA 400 - od II kw. 1987 r.

√ pakiet **mim 100**
mikroprocesorowa jednostka centralna z mikroprocesorem Z 80, RAM 64 kB i systemem operacyjnym CP/M do minikomputerów MERA 100.
* dostawa od I kw. 1987 r.

√ usługi software'owe:
oprogramowanie użytkowe do naszych mikrokomputerów.

Gwarantujemy wysoką jakość usług.
Zamówienia prosimy składać w Dziale Handlowym.
Dostawy w kolejności złożonych zamówień.



Napisz lub zadzwoń - pomożemy Ci wybrać!

GIEŁDA (notowania z 5.09.86)

		GIEŁDA BAJTKA (tys. zł)	KOMIS (tys. zł)	AUSTRIA (średnie) (öS)	FRANCJA (średnie) (FF)	RFN (średnie) (DM)	WLK. BRYT. (średnie) (£)
SINCLAIR	ZX 81	20–35	25–40	600	—	75–90	—
	ZX Spectrum 48 kB	50–80	90	1600	—	150–250	45–65
	ZX Spectrum Plus	95	140–160	1990	1400	200–300	70–90
	ZX Spectrum 128 kB	190	—	—	2200	520–590	150
	Drukarka SEIKOSHA GP50S	65–70	100	—	—	260	60–65
	Interface Kempston	8–16	20	250	300	29	7–9
	Stacja dyskietek OPUS 1(3,5")	150	—	5000	—	410	95–105
COMMODORE	C-64	130–150	150–160	3600	1900	370–450	90–110
	C-128	265–300	450	6900	2890	650–700	210–230
	C-128 D	600	—	13900	6850	1300	390–410
	Magnetofon 1531	30–35	35	900	350	80–90	30
	Stacja dyskietek 1541	155–170	200	4690	1950	470	110–150
	Stacja dyskietek 1570	210	—	6900	2300	490–540	160–170
	Drukarka MPS 801	120–130	150	2900	2200	199	—
	Drukarka MPS 803	160	200	—	2800	300–330	140–150
	Dyskietki 5 1/4 (średnia jak.)	0.7–1.8	1–2	16–25	7–18	1–5	0.8–2
ATARI	800 XL	65–80	90–100	1500	900	140–180	60
	130 XE	130–140	190	2100	1400	330	110
	Stacja dyskietek 1050	130–140	160–200	2200	2150	395	130
	Magnetofon	25–30	30–35	800	—	75	20
	Drukarka 1029	140	200	1990	—	450	85
AMSTRAD	464 z mon. monochromat.	190–210	300	6500	2690	598	160–180
	464 z monitorem kolor.	290	350–400	10000	3990	950	260–268
	6128 z mon. monochromat.	360–390	700–800	13000	3990	998	250–260
	6128 z monitorem kolor.	485	1 mln	17000	5290	1600	340–350
	PCW 8256	800	1.3 mln	—	5920	1750	380–400
	Dyskietki 3"	4–5	6	—	35	10–15	3.5–4
	Lightpen do 464	35	—	—	260	65	20

NIE TYLKO KOMPUTERY

KRYPTONIM „FOBOS"

Dokończenie ze str. 32

eksperymentów. Planuje się m.in. wykonać zdjęcia całej powierzchni w podczerwieni, co da możliwość sporządzenia mapy termicznej Fobosa i pozwoli uzyskać rozeznanie w dynamice stosunków cieplnych.

Projekt przewiduje też dostarczenie na Fobosa autonomicznej stacji, mogącej pracować w długim okresie czasu. Oddzieli się ona od statku w kulminacyjnym momencie misji, w ten sposób aby opadała wirując jednocześnie wokół swej osi (chodzi o to, aby wylądowała podstawą do dołu). Do powierzchni Fobosa „zacumowany" zostanie lądownik automatycznie za pomocą specjalnej sondy, która wbije się w grunt po odpaleniu ładunku wybuchowego. Lądownik ten zostanie m.in. wyposażony w kamerę telewizyjną o wysokiej rozdzielczości. W zestawie aparatury znajdą się przyrządy do badania składu chemicznego gruntu, jego właściwości fizycznych i inne.

Inny lądownik zrzucony na Fobosa będzie miał możliwość przemieszczania się po jego powierzchni. Za pomocą specjalnego mechanizmu będzie „odbijał się" od powierzchni i wykonywał skoki na odległości nawet dziesiątków metrów. Taka „żabia" technika poruszania się jest bardzo wygodna w przypadku ciał kosmicznych z niską grawitacją, a do takich właśnie należy Fobos.

PRZYRZĄDY XXI WIEKU

Specjaliści są zgodni, że historia kosmonautyki nie znała jeszcze tak wielofunkcyjnego i skomplikowanego statku międzyplanetarnego. Gdy akademikowi **Roaldowi Sagdiejewowi** zadano pytanie: **„A nie boicie się rozpoczynać tak skomplikowanego programu?"** szef **Instytutu Badań Kosmicznych** odpowiedział: **„Czym bardziej skomplikowany, tym bardziej interesujący. Najważniejsze, że współczesny poziom rozwoju naszej techniki pozwala nie tylko na zaplanowanie, ale i na urzeczywistnienie takiego projektu. Oczywiście, niezbędny jest olbrzymi wysiłek, przecież aparatura, która znajduje się na pokładzie nie ma analogów ..."**

Na zakończenie spotkania z **Wiaczesławem Balębanowem** zadałem mu pytanie: **— W jakim stadium znajduje się obecnie projekt „Fobos"?**

— Przygotowania idą pełną parą. W radzieckiej stoczni kosmicznej powstaje sama stacja, a w wielu laboratoriach świata trwa przygotowanie przyrządów, które znajdą się na jej pokładzie. I sama stacja, i przyrządy wykonywane są w kilku egzemplarzach, jako że zasadą radzieckiego programu kosmicznego jest zawsze wysyłanie, w kilkudniowych odstępach, dwóch statków. Tak jak do Komety Halleya leciały „Wega 1" i „Wega 2", również i w kierunku „czerwonej planety" polecą „Fobos 1" i „Fobos 2".

Aparatura do misji „Fobosów" przygotowywana jest wspólnym wysiłkiem specjalistów z krajów socjalistycznych zrzeszonych w **„Interkosmosie"** (w tym znaczny udział mają uczeni polscy), a także z Austrii, RFN, Francji, Szwecji i Europejskiej Agencji Kosmicznej. Biorąc pod uwagę wymogi miniaturyzacji, czułości i niezawodności działania już dziś wiadomo, że konstruowanie tych przyrządów związane będzie nie tylko z zastosowaniem najnowszych idei i technologii, ale zmusi uczonych i inżynierów do pokonania kolejnych barier stojących przed techniką. Będą to więc prawdziwe „przyrządy XXI wieku".

A data startu do tej niezwykłej misji? Jest już znana — nastąpi on w połowie 1988 roku.

Waldemar Siwiński

ALIBABA



Cześć Maluchy!

Znacie bajkę o Alibabie i czterdziestu rozbójnikach? Znacie! A więc zabawmy się razem.

Pamiętacie na pewno, w jaki sposób sprytni zbójcy strzegli swych skarbów. Mieli kryjówkę w skale, która otwierała się tylko na odpowiednie hasło. (Pamiętacie jakie?) Hasło znał tylko herszt bandy i tylko on — do czasu aż w całą sprawę nie wmieszał się jeszcze sprytniejszy Alibaba — mógł otworzyć skarbiec.

A może wy także macie jakieś skarby do ukrycia, może na przykład jakiś specjalny program, którego nie chcecie pokazać kolegom. Jest na to sposób. Wystarczy umieścić przed waszym programem przedstawiony tutaj podprogram. Pyta on o hasło i przechodzi do wykonywania właściwego programu tylko w przypadku podania prawidłowej odpowiedzi.

Popatrzmy jak skonstruowany jest i jak działa nasz podprogram. Skorzystamy przy tym ze schematu blokowego. Sądzę, że po kilku ostatnich odcinkach naszego cyklu każdy przedszkolak wie bardzo dobrze co to jest schemat blokowy, a nowicjusze szybko zorientują się o co chodzi.

Pierwsze dwie linie (100 i 110) to tzw. inicjalizacja programu — rozpoczęcie. Zwykle podawane są tu parametry ekranu (kolory, tryby drukowania itd.) my ograniczymy się do określenia wartości początkowych zmiennych: **haslo$:=""** (oznacza to, że zmienna tekstowa o nazwie **haslo$** jest tekstem pustym — w cudzysłowach nic się nie znajduje), oraz: **nrlitery:=0** (czyli, że zmienna liczbowa **nrlitery** ma wartość 0).

Zwyczaj oznaczania zmiennych całymi słowami jest bardzo przydatny, zwłaszcza w programach, w których występuje większa ich ilość. Po prostu łatwiej się w tym całym bałaganie połapać. W naszym przypadku zmienna **haslo$** oznacza właśnie hasło wpisywane przez usiłującego uruchomić program. Zmienna **nrlitery** oznacza natomiast kolejny numer litery wpisywanej z klawiatury (komputer w tym programie będzie odczytywał kolejne znaki a nie całe słowo).

W linii 120 komputer drukuje na ekranie komunikat: **„Podaj hasło"**, a w linii 130 sprawdza jaki klucz klawiatury został wciśnięty i odpowiedni znak podstawia pod zmienną tekstową **litera$.** Jeśli nie został wciśnięty żaden z kluczy **litera$** będzie tekstem pustym — " " i instrukcja warunkowa z linii 140 skieruje program z powrotem do linii 130. I tak w kółko, aż do chwili gdy któryś z kluczy nie zostanie wciśnięty.

Jeśli wciśniemy teraz np. literę „s" komputer przechodzi do wykonania trzech kolejnych operacji (150-170). Najpierw dodaje do zmiennej **haslo$** odczytany znak (na początku haslo$ było tekstem pustym, a więc teraz będzie oznaczało **„s"**, następnie **nrlitery** zwiększa o 1. Teraz drukuje na ekranie znak ✱, jest to informacja, że została wpisana jedna litera.

Oczywiście można kazać komputerowi drukować tę literę, która została wciśnięta, ale zwykle nie robi się tego. Chodzi o to, że ktoś mógłby podglądnąć nam przez ramię tajne hasło.

Po wykonaniu tych operacji następuje sprawdzenie wartości **nrlitery,** czyli liczby wczytanych liter (120). Jeśli będzie ona mniejsza od 5 — z tylu liter składa się nasze hasło — wracamy do wczytywania (130), w przeciwnym przypadku hasło wpisane z klawiatury porównywane jest z wzorcem (200). Każda niezgodność powoduje komunikat **„OSZUKUJESZ!!!"** i powrót do początku programu. Jeżeli natomiast wszystko się zgadza, komputer melduje **„Hasło prawidłowe"** i może zacząć właściwy program.

Ten program może być zrealizowany na każdym komputerze. W przypadku Commodore należy jedynie zmienić instrukcję **INKEY$** w linii 130 na **GET**

130 GET LITERA$

natomiast Spectrum nie przyjmuje wieloliterowych nazw zmiennych tekstowych. Trzeba więc zastąpić je nazwami jednoliterowymi, np.:

haslo$ — h$
litera$ — l$

Stosowanie polskich liter (np. ł) w komentarzach drukowanych przez komputer to już nieco trudniejsza sprawa i niestety, na każdym mikrokomputerze robi się to inaczej. Takie programy pokazujemy i nadal będziemy pokazywać w „Bajtku". Jeśli jednak — na razie — nie potrafisz tego zrobić nie przejmuj się. Program będzie działał i bez tego.

Takie zabezpieczenie programu jest oczywiście raczej żartem niż prawdziwą ochroną przed jego uruchomieniem. Każdy, kto choć trochę zna komputer potrafi wydrukować tekst programu na ekran, odczytać hasło, lub znaleźć linię, do której trzeba skoczyć, aby ominąć to zabezpieczenie. Chyba, że spowodujemy samoczynne uruchomienie programu i dodatkowo uniemożliwimy jego zatrzymanie. To jednak również zależy od rodzaju komputera.

Z drugiej strony, choć byśmy zastosowali dziesiątki innych, najbardziej nawet skomplikowanych zabezpieczeń, zawsze znajdzie się taki Alibaba, który znajdzie drogę do naszego skarbu. To tylko sprawa wiedzy, czasu i cierpliwości.

Romek

START
haslo$:="" nrlitery:=0 100-110
PISZ Podaj hasło 120
---> /litera$:=INKEY$/ 130
? litera$="" 140 — T
haslo$:=haslo$+litera$ nrlitery:=nrlitery+1 PISZ ✱ (w jednej linii) 150-170
? nrlitery<5 180 — T
nowa linia 190
? haslo$<>"sezam" 200 — T
PISZ Oszukujesz 220
PISZ Hasło prawidłowe 240
P R O G R A M

Hej!

Romku, piszę w sprawie programu zamieszczonego w „Bajtku" nr 3/4. Program ten popełnia duży błąd, mianowicie jeżeli wprowadzić imię Kuba w linii 20, to w linii 100 komputer wydrukuje

```
100 LET haslo$=""
110 LET nrlitery=0
120 PRINT "Podaj hasło"
130 LET litera$=INKEY$
140 IF litera$="" THEN GOTO 130
150 LET haslo$=haslo$+litera$
160 LET nrlitery=nrlitery+1
170 PRINT "*";
180 IF nrlitery<5 THEN GOTO 130
190 PRINT
200 IF haslo$<>"sezam" THEN GOTO 220
210 GOTO 240
220 PRINT "OSZUKUJESZ!!!"
230 GOTO 100
240 PRINT "Hasło prawidłowe"
250 REM ************************************
260 REM ****** P R O G R A M ******
270 REM ************************************
```

Historia kosmonautyki nie znała jeszcze tak wielofunkcyjnego i skomplikowanego statku międzyplanetarnego ...

KRYPTONIM „FOBOS"

Nie opadły jeszcze emocje wywołane lotem stacji „Wega 1" i „Wega 2" do Komety Halleya, a już w Instytucie Badań Kosmicznych w Moskwie ujawnione zostały szczegóły wprost porażającego wyobraźnię nowego projektu kosmicznego o nazwie „Fobos".

SATELITA PEŁEN TAJEMNIC

Fobos — to jeden z dwóch księżyców Marsa. Szczególnie głośno było o tym księżycu w latach sześćdziesiątych. Wybitny radziecki astrofizyk **Josif Szkłowski** stwierdził wówczas: **„Jeśli obliczenia są prawidłowe, to Fobos jest niewątpliwie satelitą o sztucznym pochodzeniu"**. Tak bardzo chciano aby to była prawda, że niewiele osób zwróciło uwagę na użyte przez **Szkłowskiego** słowo „jeśli", tylko większość pamiętała o drugiej części zdania. Czyżby — zadawano sobie pytanie — rzeczywiście udało się nieznanym „kosmicznym braciom" umieścić na orbicie wokółmarsjańskiej tak gigantycznego sztucznego satelitę? A **Szkłowski** uśmiechał się tylko i z właściwą sobie skłonnością do paradoksów powtarzał:

— **Opierałem się na obliczeniach astrofizyków, a one świadczą, że Fobos jest w środku pusty, a to znaczy — sztuczny!**

Nikt kategorycznie nie stwierdził wówczas, że na Marsie nie może być takiej lub innej formy życia. Więcej, w programy badań wysyłanych w kierunku „czerwonej planety" statków kosmicznych włączano niezmiennie eksperymenty mające na celu zidentyfikowanie przejawów życia.

Dzisiaj, po serii lotów radzieckich **„Marsów"** i amerykańskich **„Marinerów"** i **„Vikingów"** wiemy już, że wszystkie te eksperymenty dały wynik negatywny. Z marzeniem o spotkaniu Marsjan przyszło się pożegnać.

Ale Mars i jego księżyce w dalszym ciągu posiadają tyle tajemnic i zagadek, że ich poznanie będzie już wkrótce celem jednej z najbardziej skomplikowanych misji kosmicznych zaplanowanych przez człowieka.

NURKOWANIE W KOSMICZNEJ PRÓŻNI

Będąc w Moskwie, o wyjaśnienie szczegółów **Projektu „Fobos"** poprosiłem **Wiaczesława Balebanowa** — wicedyrektora **Instytutu Badań Kosmicznych Akademii Nauk ZSRR**.

— **Jest to program wielozadaniowy, jak większość naszych programów** — wyjaśnia **W. Balebanow**. — **Chcemy za jednym zamachem dokonać badań Słońca, badań przestrzeni kosmicznej, badań Marsa oraz — jako uwieńczenie misji — przeprowadzimy badania Fobosa. Po 200-dniowym locie, w rezultacie różnych skomplikowanych manewrów, wejdziemy na orbitę wokółmarsjańską, synchroniczną z orbitą Fobosa. Nasza stacja znajdować się będzie wówczas ok. 30 km nad powierzchnią księżyca. W kulminacyjnym momencie misji nasza stacja lotem „nurkowym" zbliży się do satelity i „zawiśnie" 50 metrów nad jego powierzchnią! Z tej wysokości, kilkunastopiętrowego wieżowca, będziemy przeprowadzać obserwacje satelity i wykonywać eksperymenty, które pozwolą nam zbadać skład i budowę Fobosa. Wnętrze satelity zbadamy na przykład za pomocą fal radiowych o dużej długości; sprawdzimy, czy rzeczywiście jest on pusty w środku ... Potem znów uniesiemy się w górę i będziemy prowadzić inne badania. Mars będzie wówczas po przeciwnej stronie Słońca niż Ziemia. Da nam to więc unikalną możliwość do równoczesnego badania — z Ziemi i z krążącej dookoła Marsa stacji — obu stron naszej gwiazdy. Zobaczymy procesy zachodzące na Słońcu w całej ich dynamice i złożoności. A zbadanie słoneczno-ziemskich więzi to przecież jedno z najważniejszych zadań naukowych naszych czasów.**

MOMENT KULMINACYJNY

Wróćmy teraz do najważniejszego i najbardziej fascynującego momentu misji. Wyobraźmy sobie, że w wyniku „nurkowego" manewru schodzenia stacja „zawisła" 50 m nad Fobosem (według najnowszych poglądów jest to najprawdopodobniej asteroida przechwycona przez pole grawitacyjne Marsa). Co się tam wówczas, 78 mln kilometrów od Ziemi, będzie działo?

Otóż, na przykład, maleńki kawałek powierzchni Fobosa, nie większy niż 1 mm^2, zostanie „ostrzelany" promieniem lasera. Na ten skrawek fobosjańskiego gruntu padnie strumień energii o mocy rzędu dziesiątek milionów watów. Impuls ten będzie oczywiście trwał bardzo krótko ale wystarczy to, aby część materii wyparowała. Wówczas zainstalowany na pokładzie stacji analizator zarejestruje i zbada te cząstki, które wyparują w stronę statku. W czasie jednego „ostrzelania" planuje się zarejestrowanie około miliona cząstek. **„Już sama idea wykorzystania takiej techniki jest tak nowatorska, że poraża umysły"** — stwierdził jeden z amerykańskich planetologów, po zapoznaniu się z radzieckimi przygotowaniami.

— **Nasz laser jest urządzeniem przeznaczonym wyłącznie do celów naukowych** — wyjaśnia kierownik naukowy eksperymentu, doktor **Georgij Managadze**. — **Metodyka drugiego aktywnego eksperymentu, jaki planujemy, oparta jest na wyemitowaniu z umieszczonego na pokładzie stacji miotacza plazmy strumienia jonów kryptonu. Strumień ten dotrze tylko do molekuł znajdujących się na samej powierzchni księżyca. Powstałe wówczas cząstki wtórne zostaną zbadane po dotarciu ich do pokładu stacji. Pozwoli to poznać skład najbardziej zewnętrznej warstwy gruntu o grubości milionowych części milimetra. Oprócz tego, być może, uda się nam zidentyfikować jony wybite z powierzchni Fobosa przez cząstki wiatru słonecznego.**

Jedną z głównych cech charakterystycznych badań Fobosa, będzie kompleksowy charakter

Dokończenie na str. 30